

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису  
УДК 519.254

До захисту допущено  
В. о. завідувача кафедри ММСА  
О.Л.Тимощук  
«\_\_\_» \_\_\_\_\_ 2019 р.

## Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз  
на тему: «Прогноз курсу криптовалюти Bitcoin на основі мереж Байєса»

Виконав:  
студент II курсу, групи КА-82 мп  
Макогон Роман Олександрович

\_\_\_\_\_

Керівник:  
доцент,  
к.т.н., Терентьев О. М.

\_\_\_\_\_

Рецензент:  
к.ф.-м.н., доцент,  
доцент кафедри прикладних  
інформаційних систем  
Київського національного  
університету імені Тараса Шевченка  
В.М. Домрачев

\_\_\_\_\_

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів  
без відповідних посилань

Студент \_\_\_\_\_

Київ  
2019

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)  
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ  
В. о. завідувача кафедри ММСА  
О. Л. Тимощук  
«\_\_\_» \_\_\_\_\_ 2019 р.

### ЗАВДАННЯ

на магістерську дисертацію студенту Макогону Роману Олександровичу

**1. Тема дисертації:** «Прогноз курсу криптовалюти Біткоїн на основі мереж Байєса», науковий керівник дисертації Терентьев Олександр Миколайович, к.т.н., доцент, затверджені наказом по університету від «08» листопада 2019 № 3862с

**2. Термін подання студентом дисертації:** 13 грудня 2019 р.

**3. Об'єкт дослідження:** Криптовалюта Біткоїн, принципи її функціонування та ціноутворення

**4. Предмет дослідження:** методи прогнозування курсу криптовалют, які використовують мережі Байєса

**5. Перелік завдань, які потрібно розробити:**

1) дослідити сучасний стан та особливості застосування мереж Байєса для прогнозування курсу криптовалют;

2) реалізувати евристичний метод побудови байєсівської мережі з використанням значення взаємної інформації (ЗВІ) та функції опису мінімальної довжини (ОМД);

3) на його основі створити програмний продукт;

4) пошук даних для застосування в програмі;

5) реалізувати метод прогнозування курсу криптовалют на основі байєсівської мережі;

6) розробити стартап-проект виведення на ринок результатів дослідження;

7) розробити концептуальні висновки за результатами наукового дослідження

**6. Орієнтовний перелік графічного (ілюстративного) матеріалу:**

1). Схема побудованої мережі Байєса (рис.);

- 2). Приклади функціонування створеного програмного продукту (рис.);
- 3). Таблиці у розділі стартап-проекту

### 7. Орієнтовний перелік публікацій:

Свідоцтво про авторське право на корисну модель “Система підтримки прийняття рішень з визначення кредитоспроможності фізичних осіб з використанням мереж Байєса” в УкрПатент.

**8. Дата видачі завдання:** 05 вересня 2019 р.

### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1.	Концептуальний вступ дисертації. Формулювання об’єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	05.09.2019—13.09.2019
2.	Перший розділ. Огляд літературно-інформаційних джерел. Понятійно-категоріальний апарат. Характеристика об’єкта	14.09.2019—25.09.2019
3.	Другий розділ. Розробка математичної моделі для задачі прогнозу курсу Біткоїн з допомогою байєсових мереж	26.09.2019—10.10.2019
4.	Третій розділ. Розробка програмного забезпечення. Аналіз перспектив комерціалізації.	11.10.2019—25.10.2019
5.	Четвертий розділ. Розробка стартап-проекту.	26.10.2019—15.11.2019
6.	Оформлення результатів дослідження та роботи програми.	16.11.2019—21.11.2019
7.	Концептуальні висновки. Перспективи розвитку отриманих рішень	22.11.2019—26.11.2019

Студент

Р.О. Макогон

Науковий керівник дисертації

О.М. Терентьєв

## РЕФЕРАТ

Магістерська дисертація: 119с., 32 табл., 28 рис., 3 дод. та 58 джерел.

КРИПТОВАЛЮТА, БІТКОІН, ПРОГНОЗУВАННЯ, ЕВРІСТИЧНИЙ МЕТОД, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ЙМОВІРІСНИЙ ВИСНОВОК, МЕРЕЖА БАЙЄСА, НАВЧАЛЬНІ ДАНІ.

У світі працюють ефективні рішення побудовані на базі Блокчейн, такі як наприклад Біткоін - інноваційна мережа платежів та цифрова валюта. З цією технологією тісно пов'язані великі дані. Одна з галузей, що зараз активно накопичує такі дані – біржі обміну криптовалют – Binance, EXMO, Bittrex, та інші. Наразі існує потреба в інструментах, які дозволяють аналізувати такі дані, щоб отримувати корисну інформацію, придатну для комерційного застосування. Мережі Байєса є ефективним інструментом для вирішення цієї задачі.

Метою дослідження є створення системи підтримки прийняття рішень (СППР), що призначена для побудови структури дискретної мережі Байєса (МБ) за навчальними даними, ймовірнісного висновку в ній, та використання її для прогнозу курсу криптовалюти Біткоін.

Досягнення поставленої мети вимагає розв'язання таких задач:

1. Аналіз принципів функціонування та ціноутворення криптовалюти Біткоін
2. Аналіз існуючих методів прогнозування курсу Біткоін.
3. Аналіз принципів побудови МБ за навчальними даними.
4. Аналіз методів побудови ймовірнісного висновку в МБ.
5. Реалізація евристичного алгоритму побудови МБ у вигляді програмного продукту.
6. Реалізація ймовірнісного висновку в побудованій мережі.
7. Апробація програмного продукту на навчальних даних, що становлять вибірку історичних даних коливання курсу криптовалюти Біткоін.

8. Аналіз ефективності створеної моделі, порівняння точності побудованого прогнозу з результатами інших досліджень у сфері прогнозування курсу криптовалют.

Об'єктом дослідження є статистичні дані щодо історії коливання курсу Біткоїн, а також 24 функцій ринку та мережі Біткоїн, зібраних щоденно з 2009 року, які потребують ефективної аналітичної обробки з метою прогнозування подальших коливань його курсу.

Предметом дослідження є метод прогнозування курсу криптовалют з використанням МБ, навчання ймовірнісних МБ за статистичними даними та формування ймовірнісного висновку.

Методи дослідження базуються на поняттях теорії ймовірності та математичної статистики, теорії графів та теорії формування ймовірнісного висновку у МБ, принципах функціонування та ціноутворення в криптовалюті Біткоїн.

Наукова новизна одержаних результатів: реалізовано евристичний метод побудови МБ за навчальними даними. Реалізовано ймовірнісний висновок в побудованій мережі. Реалізовано алгоритм виключення змінних для знаходження ймовірнісного висновку в побудованій мережі. Реалізовано СППР, що використовує розроблений метод для прогнозування курсу криптовалюти Біткоїн на різних часових проміжках. Проведено порівняльний аналіз точності прогнозування з результатами інших досліджень в сфері прогнозування курсу Біткоїн.

## **ABSTRACT**

Master thesis: 119 p., 32 tables, 28 fig., 3 add. 58 references.

CRYPTOCURRENCY, BITCOIN, FORECASTING, HEURISTIC METHOD, DIGITAL ANALYSIS, INFERENCE, BAYESIAN NETWORK, TRAINING DATA.

Relevance of the topic – effective Blockchain-based solutions exist in the world, such as Bitcoin, an innovative payment network and digital currency. Big data is closely linked to this technology. One of the industries that is now actively accumulating such data – is a cryptocurrency exchange business - Binance, EXMO, Bittrex, etc. There is now a need for tools that allow to analyze such data in order to obtain useful information for commercial use. The Bayesian network (BN) is a very effective tool for this task.

Achieving the goal requires solving the following tasks:

1. Analysis of the principles of functioning and pricing of Bitcoin cryptocurrency
2. Analysis of different methods of forecasting the Bitcoin rate.
3. Analysis of the principles of building BN based on training data.
4. Analysis of methods for constructing probabilistic inference in BN.
5. Implementation of the heuristic algorithm of BN construction as a software product.
6. Implementation of probabilistic inference in the constructed network.
7. Testing the software on training data, which is a sample of a historical data on the fluctuation of the Bitcoin cryptocurrency.
8. Analysis of the effectiveness of the created model, comparison of the accuracy of the constructed forecast with the results of other studies in the field of cryptocurrency forecasting.

Objective of the study is the statistics on the history of Bitcoin exchange rate fluctuations that require effective analytical processing to predict its future fluctuations.

The subject of the research is the method of predicting the rate of cryptocurrencies using BN, training probabilistic BN based on training data and forming a probabilistic inference.

Research methods are based on the understanding of theory of probabilities and mathematical statistics, theory of graphs and theory of inference in BN, principles of Bitcoin cryptocurrency exchange rate fluctuations.

Scientific relevance of the received results: a heuristic method of building a BN based on training data is implemented. The probabilistic inference in the constructed network is done. The algorithm of variables exclusion for probabilistic inference in the constructed network is implemented. The system is used to predict the Bitcoin cryptocurrency exchange rate fluctuations at different time intervals. A comparative analysis of forecasting accuracy is performed with the results of other studies in the field of Bitcoin exchange rate prediction.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ.....	10
ВСТУП .....	11
РОЗДІЛ 1 ОСОБЛИВОСТІ ТЕХНОЛОГІЇ БЛОКЧЕЙН ТА КРИПТОВАЛЮТИ БІТКОІН .....	13
1.1 Розподілена система управління даними Блокчейн .....	13
1.2 Блокова структура технології блокчейн .....	15
1.3 Цифрова криптовалюта Біткоїн як фінансовий інструмент .....	19
1.4 Особливості обробки транзакцій системою Біткоїн .....	22
1.5 Методи прогнозування курсу криптовалют .....	25
1.5.1 Метод на основі пошуку K найближчих сусідів .....	26
1.5.2 Метод прогнозування на основі Байєсівської регресії .....	27
Висновки до розділу 1 .....	30
РОЗДІЛ 2 МЕРЕЖІ БАЙЄСА .....	32
2.1 Мережі Байєса як інструмент для прогнозування курсу Біткоїн .....	32
2.2 Дослідження типів мереж Байєса .....	35
2.3 Побудова мереж Байєса.....	41
2.3.1 Випадок відомої структури та повних спостережень .....	45
2.3.2 Випадок невідомої структури й повних спостережень.....	47
2.3.3 Евристичний метод побудови Мереж Байєса .....	49
2.4 Алгоритми імовірнісного висновку .....	53
2.4.1 Алгоритм розповсюдження повідомлення для однозв'язних мереж (Перла).....	55
2.4.2 Алгоритм визначеного перетину.....	58
2.4.3 Алгоритм виключення змінних .....	59
Висновки до розділу 2 .....	61
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГНОЗУВАННЯ КУРСУ БІТКОІН.....	63
3.1 Опис програмного продукту .....	63
3.2 Інструкція з експлуатації програмного продукту.....	68
3.3 Результати тестування програмного продукту .....	73



Висновки до розділу 3 .....	79
РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....	81
4.1 Опис ідеї проекту .....	81
4.2 Технологічний аудит ідеї проекту.....	82
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	83
4.4 Розроблення ринкової стратегії проекту .....	89
4.5 Розроблення маркетингової програми стартап-проекту.....	91
Висновки до розділу 4 .....	96
ВИСНОВКИ.....	97
ПЕРЕЛІК ПОСИЛАНЬ .....	99
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ .....	104
ДОДАТОК Б РОБОЧИЙ ДАТАСЕТ .....	117
ДОДАТОК В ЗАЯВКА НА АВТОРСЬКЕ СВДОЦТВО.....	118

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

БД - база даних

Гбайт - гігабайт або 1024 мегабайт

ЗВІ - значення взаємної інформації

МБ - мережа Байєса

ОМД - функція опису мінімальної довжини

ПЗ - програмне забезпечення

СППР – система підтримки прийняття рішень

ТУЙ - таблиця умовних ймовірностей

BN (англ. Bayesian Network) - мережа Байєса

BTC (англ. Bitcoin) - пірінгова платіжна система, яка використовує однойменну одиницю для обліку операцій

ECDSA (англ. Elliptic Curve Digital Signature Algorithm) - алгоритм з відкритим ключем для створення цифрового підпису

ЕМ (англ. Expectation-maximization) - алгоритм в матстатистиці для знаходження оцінок параметрів імовірнісних моделей

ETH (англ. Ethereum) - платформа на базі блокчейна

IDE (англ. Integrated Development Environment) - інтегроване середовище розробки

MAP explanation (англ. Maximum a Posteriori) - максимальне апостеріорне тлумачення

MDL (англ. Minimum description length) - функція опису мінімальної довжини

MI (англ. mutual information) - значення взаємної інформації

MPE (англ. most probable explanation) - найбільш імовірне тлумачення

SHA-256 (англ. Secure Hash Algorithm) - одностороння хеш-функція, одна з функцій безпечного алгоритму хешування

## ВСТУП

У світі працюють ефективні рішення побудовані на базі Блокчейн, такі як наприклад Біткоїн - інноваційна мережа платежів та цифрова валюта. Сучасний ринок цифрових валют наразі швидко розвивається, так максимальна капіталізація ринку криптовалют донедавна складала 190 млрд. доларів, а курс біткоіна 15 грудня 2017 року досягнув 19 783 доларів США. З цією технологією тісно пов'язані великі дані. Одна з галузей, що зараз активно накопичує такі дані – біржі обміну криптовалют – Binance, EXMO, Bittrex, та інші. Наразі існує потреба в інструментах, які дозволяють аналізувати такі дані, щоб отримувати корисну інформацію, придатну для комерційного застосування, такого, як можливість прогнозування його коливань. Мережі Байєса (МБ) ефективним інструментом для вирішення цієї задачі.

Об'єктом дослідження є статистичні дані щодо історії коливання курсу Біткоїн, а також 24 функцій ринку та мережі Біткоїн, зібраних щоденно з 2009 року, які потребують ефективної аналітичної обробки з метою прогнозування подальших коливань його курсу.

Предметом дослідження є метод прогнозування курсу криптовалют з використанням мереж Байєса, навчання ймовірнісних мереж Байєса за статистичними даними та формування ймовірнісного висновку.

Метою дослідження є створення СППР, що для прогнозування курсу криптовалюти Біткоїн використовує дискретну мережу Байєса, побудовану за навчальними даними та ймовірнісний висновок у цій мережі.

Досягнення поставленої мети вимагає розв'язання таких задач:

1. Аналіз принципів функціонування та ціноутворення криптовалюти Біткоїн
2. Аналіз існуючих методів прогнозування курсу Біткоїн.
3. Аналіз принципів побудови МБ за навчальними даними.
4. Аналіз методів побудови ймовірнісного висновку в МБ.

5. Реалізація евристичного алгоритму побудови МБ у вигляді програмного продукту.

6. Реалізація ймовірнісного висновку в побудованій мережі.

7. Апробація програмного продукту на навчальних даних, що становлять вибірку історичних даних коливання курсу криптовалюти Біткоїн.

8. Аналіз ефективності створеної моделі, порівняння точності побудованого прогнозу з результатами інших досліджень у сфері прогнозування курсу криптовалют.

Інструменти розробки включають в себе елементи теорії ймовірності та математичної статистики, теорії графів та теорії формування ймовірнісного висновку у МБ, принципи функціонування та ціноутворення в криптовалюті Біткоїн. Для створення СППР була також використана мова програмування SAS Base (побудова структури мережі) та мова Python 3 (навчання параметрів мережі, побудова ймовірнісного висновку та прогнозування).

Кінцевою метою даної магістерської дисертації є написання СППР для прогнозування курсу криптовалюти Біткоїн на основі МБ за попередньо накопиченим масивом фактичних даних та показників, з метою мінімізування інвестиційних ризиків.

Інформаційна база роботи включає в себе наукові статті, літературні джерела та дослідження, розміщені онлайн, що присвячені проблемі, яка розглядається.

# **РОЗДІЛ 1 ОСОБЛИВОСТІ ТЕХНОЛОГІЇ БЛОКЧЕЙН ТА КРИПТОВАЛЮТИ БІТКОІН**

## **1.1 Розподілена система управління даними Блокчейн**

Метою дослідження, що проводиться в цій роботі є розробка методу, що дозволить прогнозувати курс криптовалюти Біткоїн, використовуючи для цього такий інструмент аналізу даних, як мережі Байєса. Але для того, щоб визначити, які дані необхідні для того, щоб на їх основі можна було б прогнозувати курс Біткоїн, потрібно розглянути принципи функціонування та ціноутворення криптовалюти Біткоїн. Біткоїн побудований з використанням технології блокчейн, розуміння якої необхідне для подальшого аналізу досліджуваної криптовалюти.

Блокчейн (Blockchain) [1] - це спеціального типу структура, призначена для запису групи транзакцій, розподілена система управління даними, де будь-який вузол цієї системи є рівноправним у доступі до них [2]. Тобто, система має горизонтальну ієрархію. Тому такі системи фактично є несприйнятливими до падінь. Звісно, їх можуть намагатися зламати, однак, щоб вивести з ладу таку розподілену систему, зловмисник має взяти під контроль або забезпечити зміну більш ніж 50% вузлів такої системи [3]. Суттєві витрати на спроби зробити це самотійно, зроблять економічно недоцільним спроби злому та нівелюють більшу частину потенційного виграшу.

Переваги розподіленої системи [3]:

1. Немає необхідності у посередниках.
2. Недоцільно ламати систему з економічної точки зору, тому система є надзвичайно надійною.
3. Повна прозорість розподіленої системи є причиною того, що, вчинення шахрайства є досить складним.

Недоліки розподіленої системи:

1. Такі системи досі вважаються новітніми, і їх технології знаходяться в процесі постійного змінення.

2. Для стабілізації таких систем необхідні значні кошти.

Технологія блокчейн використовує можливості криптографічного хешування [4], та одночасно з ним цифровий підпис з метою забезпечення та збереження транзакцій, які, потрапивши до системи, вже неможливо змінити. Ця технологія отримала багато прихиликів завдяки, насамперед, її використанню при побудові криптовалюти Біткоїн. Технологія Блокчейн – інноваційного типу «книга обліку», де кожний користувач може вносити власні та читати записи інших користувачів. Це технологія характеризується трьома функціями безпеки [5]: незмінність, прозорість і автономія. Створена на основі цієї технології програма дозволяє зберігати різноманітні дані, використовуючи мережу Інтернет захищеним і прозорим способом. При цьому взагалі не маючи центру керування.

Блокчейн за своєю структурою є ланцюгом блоків, в яких має зберігатися інформація якогось виду, до прикладу: транзакції, документи про власність, фінансові контракти, твори мистецтва, патенти. Поява і стрімкий ріст популярності Біткоїн сприяли широкому розповсюдженню технології блокчейн, на якій і будується система цифрових валют. Спосіб зберігання даних, який також називається цифровим реєстром операцій, впорядкований у блоки згідно із ланцюговим принципом та має назву – «Blockchain» (англійською block — блок, chain — ланцюг).

Блокчейн, в сенсі блоку транзакцій є спеціалізованою структурою для створення та зберігання записів групи транзакцій. Транзакції здійснюються лише коли вважаються підтвердженими. Це безпечно, надійно та зручно, коли йдеться про проведення платежів або про передачу конфіденційної інформації. З метою, щоб транзакція вважалась підтвердженою, формат її та підписи мають бути перевірені. Надалі групу транзакцій записують до спеціального блоку. В цих блоках всі дані можна швидко перевірити. У кожному наступному блоці зберігається інформація про попередній блок. Наприклад, при операціях із криптовалютами, у ланцюжку блоків записується інформація про всі виконані до цього дії з біткоїнами.

До блоку входять заголовок та список транзакцій. Заголовок має свій власний хеш-код [4], хеш-коди транзакцій, хеш-код попереднього блоку, та додаткову службову інформацію. Першою в транзакційному блоці вказується отримання комісії, яка буде нагородою тому користувачу, що створить даний блок. Деревоподібне хешування [4] використовується для проведення транзакцій в блоці.

В зв'язку з тим, що результат функції SHA-256 [6] (хешування) є непередбачуваним, крім випадкового перебору, немає алгоритму отримання необхідного результату. Якщо хеш-код не задовольняє умові, тоді змінюється блок службової інформації в заголовку і хеш-код перерахується. Вузол розсилає всім іншим підключеним вузлам, які будуть перевіряти блок, отриманий блок тоді, коли співпадають варіанти. У випадку, при якому блок не містить помилки, він вважається доданим до ланцюга і наступний блок має включати його хеш-код. Далі все знов починається спочатку.

## 1.2 Блокова структура технології блокчейн

Структура даних Блокчейн - це пов'язаний між собою список блоків транзакцій, що упорядкований «назад» [3]. Блокчейн може зберігатися у файлі або базі даних. Вираз «блоки пов'язані назад» означає, що кожен блок в ланцюзі посиляється на попередній блок. Блокчейн часто зображується як вертикальна піраміда з блоками, розташованими один поверх одного, перший блок є платформою для всіх інших блоків, що розташовані вище [7]. Подібне зображення у вигляді складених один на одного блоків є причиною для використання таких термінів, як "висота" для позначення відстані між першим блоком, і "вершиною", що вказує на доданий нещодавно блок [7].

Кожний блок в Блокчейн ідентифікується хеш-кодом [4], що згенерований за допомогою SHA256 [6] - криптографічного алгоритму, застосованого до заголовка блоку. Кожен з блоків посиляється на попередній блок-предок, використовуючи для цього хеш-код попереднього блоку в

заголовку. Кожний блок включає хеш-код власного батька всередині свого заголовку. Послідовність хеш-кодів, які зв'язують блок із своїм батьком створює ланцюжок. Цей ланцюжок тягнеться до найпершого з коли-небудь створених блоків, який є блоком генезису.

Блок має тільки одного батька. Блок може тимчасово мати кілька дочірніх блоків. Кожний із цих дочірніх блоків посилається на один і той самий батьківський блок, і поле "хеш-код попереднього блоку" містить той самий хеш-код.

Зазнає змін хеш-код дочірнього блоку у випадку, коли змінюється хеш-код батьківського блоку. У тому випадку, коли зазнає змін батьківський блок, також зазнає змін і його хеш-код. Хеш-код батьківського блоку, що зазнав змін, відповідно вимагає зміни в дочірньому блоці посилання "хеш-коду попереднього блоку". Це, відповідно, вже в свою чергу змінює хеш-код цього дочірнього блоку. Це надалі змінює посилання у свого попереднього блоку. Він, в свою чергу, змінює хеш-код власного попереднього блоку, і таким самим чином далі по ланцюгу. Каскадний ефект змін забезпечує, що в разі, якщо за блоком існувало багато поколінь, в цей блок не можуть бути внесені зміни без обов'язкового перерахунку всіх наступних за ним блоків. Так, для подібного перерахунку потрібний величезний обсяг обчислень. Чим довшим є ланцюг блоків, тим глибшу історію в блокчейні він робить незмінною, ця обставина є ключем до безпеки платформи біткоіна.

Базовою складовою одиницею блокчейна є блок. Блок представляє структуру даних, своєрідний контейнер, що об'єднує транзакції для включення в загальнодоступну бухгалтерську книгу обліку, в Блокчейн [6]. Складовими частинами блоку є заголовок (Head), що містить метадані, за заголовком іде довгий список транзакцій (Payload), (рис. 1.1.), і вони за обсягом займають більшу частину всього блоку, що наочно продемонстровано на рис. 1.1. Розмір даного блоку є 80 байт, середня транзакція займає не менш ніж 250 байтів, в середньому блок може містити більше 500 угод [8]. Отже, повністю заповнений транзакціями блок за обсягом в тисячу разів більше, ніж заголовок.



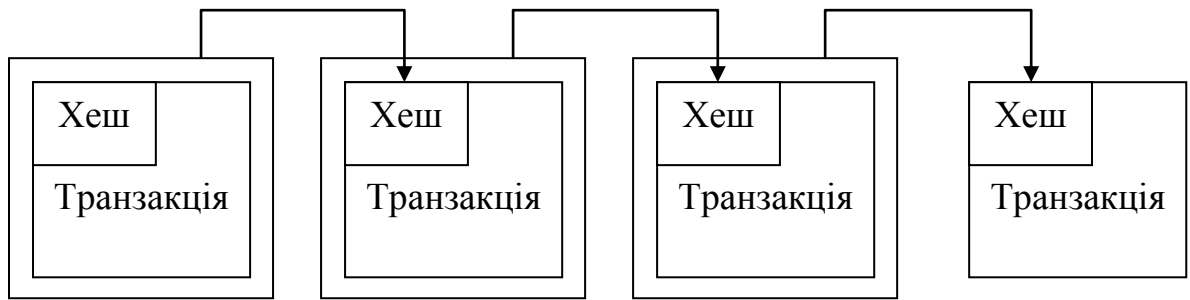


Рисунок 1.1 – Структура блоку транзакцій [8]

Блоки, що вже потрапили в блокчейн, що є важливим, змінити не можна. Так, будь-яке редагування інформації про транзакції в блокчейні заборонене. Можливе тільки дописування нових блоків.

Заголовок блока включає таку інформацію: версія блоку, дата й час створення блоку, хеш-код заголовку блока, хеш-код попереднього блоку, хеш-код усіх транзакцій в блоці, спеціальні параметри nonce та bits, що вносяться при майнінгу [8].

Хеш-код заголовку блока пов'язує попередній блок з наступним в ланцюзі блокчейна. Хеш-код прописується до наступного блоку як хеш-код попереднього блоку, і далі по ланцюжку.

До того ж в заголовку зберігається також хеш-код транзакцій поточного блоку. Він обчислюється за алгоритмом, відомим, як дерево Меркла (Merkle tree) [4], інша назва - бінарне дерево хешів [9], як видно на рис. 1.2. нижче:

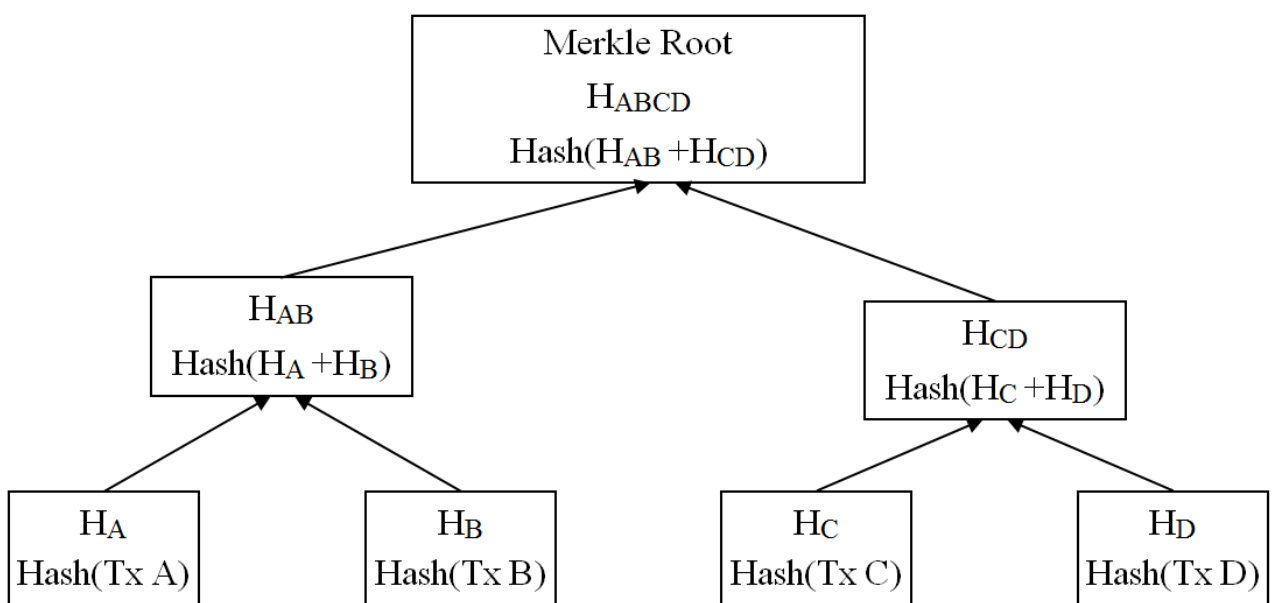


Рисунок 1.2 – Бінарне дерево хешей [9]

На першому етапі підраховуються хеш-коди всіх транзакцій. Далі обчислюється сума усіх хеш-кодів пар транзакцій, потім обчислюються хеш-коди від суми отриманих пар хеш-кодів, по тій же системі далі й далі доти, доки не отримаємо єдиний хеш-код, який буде хеш-кодом транзакцій в блоці.

Заголовки дозволяють відслідковувати цілісність вмісту відповідних блоків.

В блокчейні біткоіна записані транзакції вигляду “3 < address 1> відправник <N> біткоінів на < address 2> “

Транзакція, що записана до блоку блокчейна, насправді складніша, тому, що протокол біткоіна містить такі поняття, як Входи (In чи Inputs) і Виходи (Out чи Outputs) [9].

В цифрових валютах нові транзакції через Входи (їх може бути або один або декілька) посилаються на Виходи (їх також може бути або один або декілька) попередніх транзакцій і формують Виходи (також або один або декілька) для застосування в наступних транзакціях.

Нова транзакція С має в собі посилення на дві вхідні транзакції- А і В. На рис. 1.3 відображено, що вкінці на вході у транзакції отримаємо 0.008 BTC ( $0.005 + 0.003$ ), і вже, відповідно, надалі розділяються на два виходи - на першу адресу надсилається 0.003 BTC, на другу адресу шлеться 0.004 BTC. Залишок (0.001 BTC) формує комісію майнеру [8].

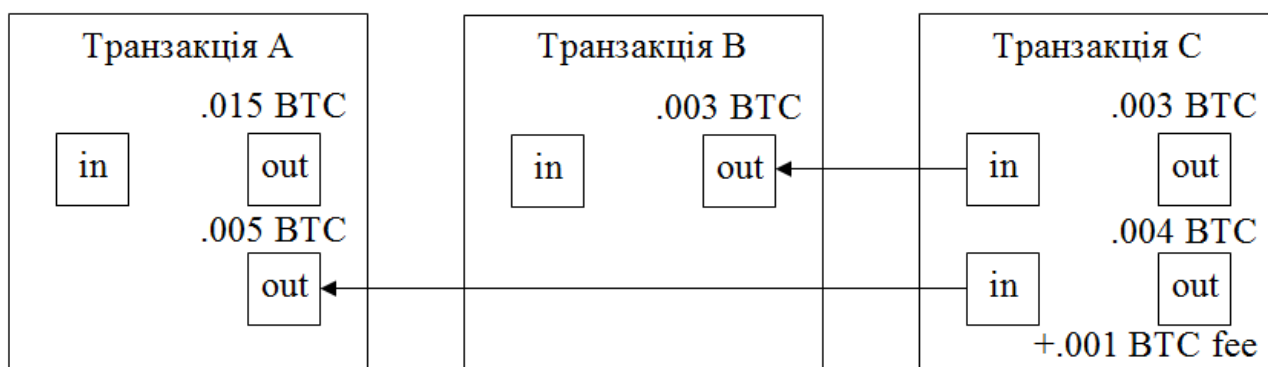


Рисунок 1.3 – Нова транзакція з посиленнями на дві вхідні транзакції [9]

Блоком генезиса називають перший блок в блокчейн [10]. Він був створений десять років тому і є спільним предком для всіх блоків в блокчейн [11]. Тому, що якщо взяти будь-який блок і простежити історію ланцюжка в

часі, то в кінцевому підсумку відлік дійде до блока генеза. Кожний вузол в блокчейн завжди починається щонайменше з одного блоку тому, що блок генезису закодований статично в клієнтському програмному забезпеченні, а це означає, що він не може бути модифікованим. Кожний вузол завжди "пам'ятає" хеш-код блоку генезису та його структуру, фіксований момент часу, коли він був створений, і єдину транзакцію в даному блоці. Тож, кожний вузол має так звану відправну точку для блокчейн, своєрідний безпечний "корінь", з якого завжди можна будувати надійний і безпечний блокчейн.

Вузли всередині містять локальну копію блокчейн, розпочинаючи з блоку генезиса. Локальна копія блокчейн безперервно оновлюється, доповнюючись з кожною появою нових блоків. Приймавши вхідні блоки з мережі, вузол виконує перевірку цих блоків, далі зв'язує з існуючим вже ланцюжком на підставі хеш-коду попереднього блока. Наприклад, вузол має кількість 277,314 блоків в копії блокчейн локально. Цей вузол «роздивляється» вміст поля `previousblockhash` цього нового блока, в якому міститься хеш-код «батьківського» блока. Цей хеш-код вузол вже знає як хеш-код останнього блоку в ланцюзі на висоті величини 277314. Саме цей новітній блок стає дочірнім блоком останнього блока в ланцюзі і розширює поточний блокчейн. Вузол доставляє цей новий блок в кінець існуючого вже ланцюга, що робить блокчейн довшим з новою висотою, що дорівнює 277315 [12].

### 1.3 Цифрова криптовалюта Біткоїн як фінансовий інструмент

Започаткована в 2009 році інновація Біткоїн (Bitcoin, BTC) [13] являється першою цифровою децентралізованою криптовалютою. Інноваційна валюта була створена всупереч всім раніше існуючим платформам електронних валют та платіжних систем. Вона з народження не прив'язана до жодних фізичних активів та «офіційних» валют, ціна цифрового біткоїна регулюється виключно попитом і пропозицією ринку. У біткоїна є одна суттєва схожість з золотом, яка

полягає в обмеженій кількості його загального запасу. Біткоїн обмежений у кількості 21 000 000 [14].

Біткоїн є всесвітньо визнаною платіжною системою, через яку можна проводити операції, без емісії та інфляції [15]. Головна перевага перед традиційними платіжними системами полягає у відсутності будь-якого єдиного керуючого і процесингового центру - всі операції на платформі біткоїна проводяться виключно в мережі рівноправних клієнтів, що є важливою особливістю.

Біткоїн історично є першою й найбільш децентралізованою криптовалютою [16]. Існують сотні інших альтернативних криптовалют (альткоїнів), наприклад, Litecoin [16] або Dogecoin [16], та все ж біткоїн має близько 90% ринкової капіталізації серед усіх криптовалют, він став стандартом де-факто.

У інноваційній предметній галузі біткоїнів та її термінах нелегко попервах зорієнтуватися, бо слова «біткоїн» і «блокчейн» можна використовувати в будь-якому з трьох сенсів [17]: як базова блокчейн-технологія, як протокол, який описує, яким чином відбуваються операції в ланцюзі блоків і клієнт, щоб мати можливість виконати транзакції, і як позначення і концепція самої криптовалюти (цифрових грошей). У блокчейн-індустрії терміни часто змішуються і відокремлення термінів відбувається по контексту, оскільки поки ще триває процес формування методологічних засад інноваційного багаторівневого стеку найсучасніших методологій.

Біткоїн є цифровими грошми, за допомогою яких купують і продають товари через Інтернет, на яких заробляють капітали, «майнячи» [18] їх або торгуючи ними на спеціалізованих біржах, створюючи інноваційне програмне забезпечення в безлічі галузей світової економіки. Таким чином, ланцюг доданої вартості біткоїнів формується кількома групами спільноти: розробниками, майнерами, біржами, сервісами обробки платежів, операторами інтернетовських гаманців і кінцевими користувачами [19]. Спочатку для роботи з криптовалютою користувач повинен мати лише біткоїн-адресу, секретний

ключ та програму-гаманець. Біткоїн-адресою є ідентифікатор, подібний до номера банківського рахунку, на який інші користувачі зможуть переказувати біткоїни. Секретним ключем є цифровий криптографічний ключ, що дозволяє здійснювати перекази наявних біткоїнів іншим користувачам. Для оперування біткоїнами, необхідно на комп'ютері або на смартфоні встановити програму-гаманець. Немає потреби відкривати розрахунковий рахунок у банку або іншій установі, процедура діє таким чином, що після установки програма автоматично генерує цифрову зв'язку секретного ключа і біткоїн-адреси, і відразу ж користувач має змогу розпоряджатися коштами, що прив'язані до даної адреси. Гаманець може мати копію блокчейна у вигляді записів усіх транзакцій, що коли-небудь були здійснені з використанням даної валюти. Це надає можливість самостійної верифікації будь-яких транзакцій в межах децентралізованої платформи Біткоїн.

Біткоїн наразі впевнено займає нішу «валюти інтернету», є глобальною загальноновизнаною цифровою платіжною системою. Історично першою і найочікуванішою галуззю застосування біткоїна стали грошові розрахунки. Наявність альтернативних грошових систем є виправданою, насамперед, з міркувань економії: зниження комісій за платежі платіжними картками в усьому світі з 3% навіть до 2% стане суттєвою перевагою для економіки. Особливо це істотно для міжнародного ринку грошових переказів, чий обсяг складає понад 500 млрд доларів щорічно, де комісії за переказ можуть сягати від 7% до 30% усієї суми [20]. Крім того, блокчейн переказує кошти миттєво, користувачеві не доводиться очікувати переказу по кілька днів.. Суттєва риса, притаманна грошовим розрахункам на основі блокчейна, в тому, що вони дозволяють забезпечувати здійснення будь-яких угод через інтернет взагалі без посередників.

## 1.4 Особливості обробки транзакцій системою Біткоїн

Факт здійснення транзакції зазначає мережі, що власник якоїсь кількості біткоїнів доручив передачу визначеної ним кількості біткоїнів іншому власнику. Вже новий власник зможе витратити ці біткоїни через створення іншої транзакції, яка далі в свою чергу дозволяє здійснити передачу вже іншому власнику біткоїнів, і далі по ланцюгу зміни власників. Транзакції аналогічні записам у звичній бухгалтерській книзі обліку доходів та витрат. Іншими словами, кожна транзакція має містити один або кілька "входів", через які надходять кошти. З іншої сторони у транзакції може бути один чи більше "виходів", куди гроші витрачаються. Транзакція має містити в собі певні докази щодо права володіння у вигляді цифрового підпису власників біткоїнівських сум на кожному зі входів. Цифрові підписи власників біткоїнів можуть бути незалежно перевірені будь-ким ще. В термінах предметної галузі біткоїнів, "витратити" означає підписати транзакцію, що переміщує гроші з якоїсь іншої попередньо здійсненої транзакції їх новому власнику, ідентифікованому за допомогою біткоїн-адреси.

Біткоїн-адреса – це є ідентифікатор (використовується в якості номеру рахунку), що має починатися з 1 або 3 і може містити 26-35 букво-цифрових символів латинського алфавіту (крім 0, O, I). Адреса також може бути записана у вигляді QR-коду, є анонімною, бо не містить інформації про свого власника. Її можна згенерувати безкоштовно, використовуючи, наприклад, програмне забезпечення платформи Біткоїн.

Вхід (input) є посиланням на вихід певної іншої транзакції. У транзакції, як правило, буває кілька входів. Значення посилань сумуються, й загальна сума біткоїнів може використовуватися у виході даної поточної транзакції. Previous tx - це є хеш-код попередньої транзакції. Index – це є окремий вихід цієї транзакції. ScriptSig – це є перша половина скрипта.

Скрипт містить два компонента: підпис та публічний ключ. Public key належить користувачу, який використовує виходи транзакції та підтверджує те,

що даний власник транзакції дійсно має право розпоряджатися сумою біткоїнів, що надходить з виходів. Інший компонент - це є ECDSA-підпис [4] хеш-коду спрощеної версії транзакції. Підпис, об'єднаний з публічним ключем, підтверджує факт, що певна транзакція була створена реальним власником даної біткоїнівської адреси.

Вихід (output) містить інструкції щодо відправлення біткоїнів. Значення (value) - це є кількість Сатоши ( $1 \text{ BTC} = 100,000,000$  Сатоши). А ScriptPubKey – це є другою половиною скрипта.

Кожний з виходів транзакції може використовуватися як вхід для вже наступної транзакції тільки єдиноразово, отже сума всіх входів для певної поточної транзакції має бути використана на її виходах.

До прикладу, якщо кількість введених на вході біткоїнів дорівнює 50 BTC, а користувачу треба відправити тільки частину з них, а саме 25 BTC, тоді біткоїн створить два вихода по 25 BTC кожний: один надішлеться в пункт призначення, а інший надішлеться ще раз власнику цих коштів (так би мовити "здача" – тобто, транзакція, яку користувач по факту відправляє собі сам).

Будь-яка сума входів біткоїнів, що не є використаною в виходах, стає комісією транзакції. Ця комісія дістанеться тому користувачеві, хто згенерує блок.

Транзакція стає елементом, що розповсюджується біткоїн-мережею. Але логіка процесу полягає в тому, що транзакція не стане записом в глобальній базі, так би мовити бухгалтерській книзі обліку (блокчейн) доти, доки не буде перевірена та включена в блок. Процес цей якраз і має назву майнінг.

Майнінг (з англ., "Mining") [18] - це є процес запису транзакцій в блокчейні. Мета майнінга – це досягнення консенсусу між вузлами мережі стосовно того, які ж транзакції будуть вважатися реальними [21].

Крім того, майнінг є єдиним способом емісії біткоїнів, що нараховуються як винагорода за рішення майнеру встановлених математичних задач за допомогою його комп'ютерного обладнання.

Кожний блок має містити підтвердження того, що завдання щодо вирішення певних математичних завдань було успішно виконано, і кожен з нодів мережі можна легко перевірити, чи насправді блок було закрито по певним правилам. Емісія цифрової валюти здійснюється в якості винагороди за видобування децентралізовано, що можна цілком стверджувати, про відсутність контролю над емітуванням зі сторони єдиного центру [22].

Процес видобутку біткоінів слугує водночас двом цілям [18]:

- Майнінг створює (емітує) нові біткоіни в кожному новому блоці, є певна аналогія з тим, як центральні банки країн друкують нові гроші.
- Майнінг створює довіру і безпеку, гарантуючи, що для включення транзакцій в блок необхідна певна обчислювальна потужність. Більше блоків вимагають більше обчислень, більше довіри.

Випуск нових біткоінів відображений на рис. 1.4, є децентралізованим, незалежним від будь-якого єдиного регулюючого й контролюючого органу, обмежений обсяг емісії відомий заздалегідь.

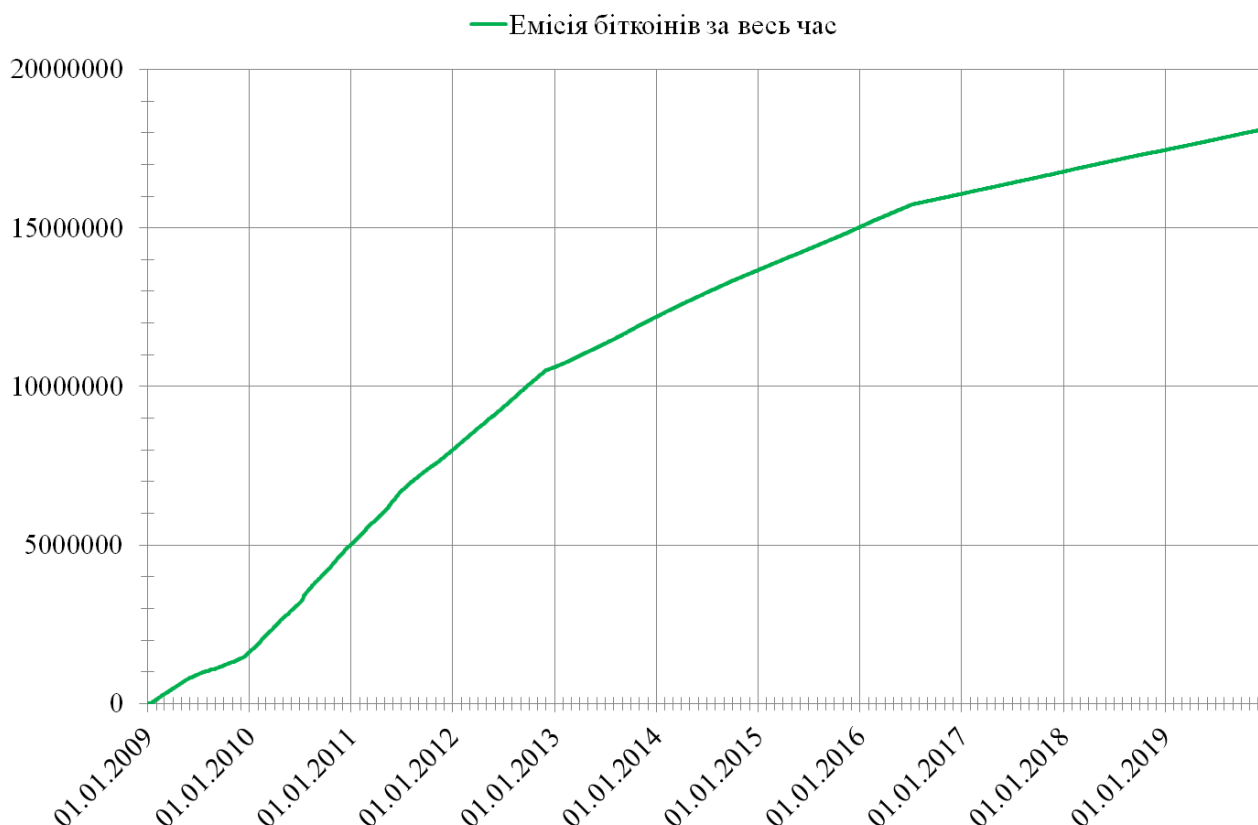


Рисунок 1.4 – Графік кількості біткоінів з 2009 року до 2019 року [22]



На ранніх етапах існування біткоінів розмір встановленої емісії при операції створення блоку дорівнював 50 біткоінів [23]. Проте 28 листопада 2012 вперше відбулось зменшення емісійної нагороди з 50 біткоінів в два рази до 25 біткоінів. Емісійна винагорода зменшилася вдруге 9 липня 2016 року знов вдвічі з 25 біткоінів до 12,5 біткоінів [23]. У майбутньому 2031 році, тобто через 12 років з цих пір розмір емісії при виконанні операції створення блоку буде скласти менше одного біткоіна і продовжить наближатися до нуля. На далеке майбутнє передбачається, що емісія зупиниться тільки в 2140 році, бо винагорода за блок не зможе більше перевищити 10-8 BTC, хоча задовго до цього моменту поступово основним джерелом винагороди платформа біткоіну за формування нових блоків надалі стануть комісійні збори [24].

### 1.5 Методи прогнозування курсу криптовалют

Далекого 2009 року, коли криптовалюта тільки-но з'явилася, її вартість дорівнювала \$0 [23]. Наступного року вартість валюти Біткоін ще не перевищувала \$0.39 [23]. Однак, вже протягом 2013 року вартість 1 Біткоін перейшла психологічну межу \$1000, а 15 грудня 2017 року вартість біткоіна досягнула 19 783 доларів США [23]. Вочевидь, для багатьох людей це мало асоціації з ринком акцій, тож обізнані у майбутніх перспективах або удачливі люди почали активно купувати криптовалюту для того, щоб заробити на купівлі-продажу. Однак подальша тенденція була такою, що в кінці 2017 року вартість біткоіна знизилась до майже \$13000 [23], що означало, що волатильність була дуже високою. Ця тема наразі не даю спокою багатьом, хто вже має у власності біткоіни, або тільки замислюється над їхньою купівлею, отже задача прогнозування курсу криптовалют є на даний час релевантною.

На те, що курс біткоіна характеризує висока волатильність, існує декілька об'єктивних причин:

- перший фактор в тому, що цифрові криптовалюти в порівнянні зі звичними валютами, які емітуються та обіг яких регулюється банківськими

системами, мають порівняно менші розміри ринку. Це значить, що навіть нібито незначні коливання криптовалют можуть мати видимий вплив на їхній курс. На додачу, розподілення багатства в галузі криптовалют іще більше нерівномірне, ніж традиційного для всіх багатства, тому окремі люди з великими частками в загальному обсязі криптовалют володіють непропорційно великою владою над курсом біткоіна;

– наступний фактор є в тому, що загальне сприйняття криптовалют (насамперед, найпопулярнішого Біткоін) є достатньо дихотомічним. До прикладу, недавні коливання вбік подорожчання можуть бути пов'язані з низкою економічних факторів (збільшення інтересу та довіри до криптовалют в усіх країнах Азії, збільшення використання підприємствами, валюта Litecoin успішно активізує SegWit для того, щоби збільшити пропускну спроможність транзакцій) [25], підіграли цікавість інвесторів та збільшили позитивні настрої в цьому просторі. З іншої сторони, об'єктивно наявні негативні зміни в законності використання криптовалют, а також збої у функціонуванні систем самих криптовалют можуть дуже легко (назавжди або на деякий час) знищити їхні суспільні переваги [26].

Задача прогнозування курсу є складною та дуже важливою внаслідок того, що криптовалюти мають високу волатильність.

#### 1.5.1 Метод на основі пошуку $K$ найближчих сусідів

Даний метод був реалізований в вигляді Twitter-бота [27], що кожні 2 години постить прогнозні значення курсу Біткоін протягом наступних  $N$  днів, де  $N$  – це є кількість днів, якою люди цікавляться найбільш. Так, до прикладу, якщо 3 людини робили запити боту спрогнозувати значення на наступні 5 днів, а 7 людей просили бота спрогнозувати значення протягом 2 наступних днів, тоді бот робив пост із прогнозом на 2 дні, оскільки більшість людей цікавило саме це передбачення.

Для побудови прогнозу бот виконував наступний алгоритм:

1. Збір всіх запитів.
2. Отримання кількості днів, яка є найбільш запитуваною для передбачення.
3. Отримання поточного значення курсу біткоїна.
4. Знаходження множини K найближчих дат за останніх 2 місяці, коли значення біткоїну було більш всього схоже на поточне значення.
5. Пошук значень BTC щодо кожної знайденої дати (дамо назву PAST\_DATE) після наступних N днів (дамо назву N\_DAYS\_AFTER\_PAST\_DATE).
6. Обчислення різниці між N\_DAYS\_AFTER\_PAST\_DATE і значенням величини PAST\_DATE щодо кожної дати.
7. Обчислення для різниць середнього значення.
8. Отриманий результат покаже наскільки біткоїн у середньому зріс за даний проміжок часу з-поміж усіх проміжків часу PAST\_DATES і проміжками часу N\_DAYS\_AFTER\_PAST\_DATES.

Цитуючи висловлення автора даного бота [27], прогноз фактично не завжди відповідає правильному значенню, хоча в переважній більшості випадків похибка може скласти близько 100-200 \$.

Недолік алгоритму в тому, що він лише аналізує історичні дані біткоїна та робить прогноз на базі фактичних минулих подій [28].

### 1.5.2 Метод прогнозування на основі Байєсівської регресії

Devavrat Shah та Kang Zhang [29] винайшов і оприлюднив метод, який використовує Байєсівську регресію, що містить використання моделі прихованого джерела.. Автори [29] користуються даними, зібраними з усіх основних бірж біткоїнів кожну секунду протягом п'яти місяців 2014 року, що в результаті становило понад 200 мільйонів точок даних з інтервалом в 10 секунд, щоби запропонувати користувачу оптимальну стратегію інвестування. Торгова стратегія є дуже простою: у кожен момент часу виконується підтримка

позиції з +1 біткоїн, або ж з 0 біткоїн або з -1 біткоїн. В кожен момент часу прогнозується середній рух курсу за наступні 10 секунд (до прикладу,  $\Delta p$ ), використовуючи методологію байєсівської регресії, і якщо  $\Delta p > t$  є порогом, тоді необхідно купувати Біткоїн, якщо ж поточна позиція  $\leq 0$ ; якщо  $\Delta p < -t$ , то вже необхідно навпаки виконувати продаж 1 біткоїну якщо значення поточної позиції  $\geq 0$ ; в протилежному випадку не треба нічого робити. Вибір часових кроків, в які приймаються торговельні рішення, що згадані вище, вибираються прискіпливо, дивлячись на останні тенденції.

Обраним методом прогнозування середньої зміни курсу  $\Delta p$  за визначений 10-секундний інтервал є байєсівська регресія. Враховуючи той факт, що часовий ряд зміни вартості Біткоїн протягом інтервалу в декілька місяців, вимірюється кожних 10 секунд, маємо значний за обсягом часовий ряд (або вектор). Використовуючи цю ретроспективу, виконується генерування трьох підмножин складових отриманого часового ряду, що має три різних довжини:  $S_1$  - 30 хвилин,  $S_2$  - 60 хвилин та  $S_3$  - 120 хвилин. У певний момент часу, аби спрогнозувати майбутню зміну  $\Delta p$ , використовуються історичні дані трьох довжин: передуючі 30 хвилин, відповідно 60 хвилин та відповідно 120 хвилин - позначені  $x_1$ ,  $x_2$  та  $x_3$ . Величину  $x_j$  використовують з історичними ретроспективними зразками  $S_j$  для прогнозування середньої зміни ціни  $\Delta p_j$  для  $1 \leq j \leq 3$  для Байєсівської регресії. Потрібно також обрахувати величину  $r = (v_{bid} - v_{ask}) / (v_{bid} + v_{ask})$ , де  $v_{bid}$  це величина що відповідає загальному обсягу, який готові придбати учасники-користувачі у найвищих 60 замовлень і  $v_{ask}$  - це загальний обсяг, який готові продавати (реалізувати) учасники-користувачі у 60 найвищих замовлень, що базується на масиві поточних даних замовлень (дані взяті з ресурсу Okcoin.com). Значення остаточної оцінки параметру  $\Delta p$  обчислюється як  $\Delta p = w_0 + \sum w_j \Delta p_j$   $3j=1 + +w_4 r$ , де обчислені  $w = (w_0, \dots, w_4)$  – параметри, що зараз у процесі навчання.

Для пошуку величин  $S$  та  $w$  весь часовий проміжок, що використовується для прогнозу, розбитий на три майже рівних періоди. Перший період використовується для знаходження шаблонів  $S_j$ ,  $1 \leq j \leq 3$ . Другий же період

використовується для вивчення параметрів  $w$ , останнім йде третій період, який призначений для визначення оцінки ефективності алгоритму. Вивчення  $w$  виконується як пошук найліпшої лінійної функції над усіма параметрами  $S_j$ ,  $1 \leq j \leq 3$ . Для вибору величини  $S_j$ ,  $1 \leq j \leq 3$  визначаються усі можливі з часових рядів відповідних довжин (вектори розмірностей 180, 360 й 720 відповідно для  $S_1$ ,  $S_2$  й  $S_3$ ). Кожна з них формує  $x_i$  та відповідна мітка  $y_i$  обчислюється як перегляд середньої величини зміни ціни в інтервалі розміром 10 секунд після завершення  $x_i$ . Шаблони поєднуються разом в 100 кластерів, використовуючи виконання алгоритму k-means. З цих 100 шаблонів відбираються 20 найбільш ефективних кластерів, далі були взяті репрезентативні шаблони, утворені з даних кластерів [30].

Зображення на рис. 1.5 демонструє результат виконання методу прогнозування, в якому синьою лінією позначений курс Біткоїн, зеленими точками позначені точки, в яких алгоритм пропонує продати біткоїн, червоними точками – коли алгоритм пропонує купити [31].

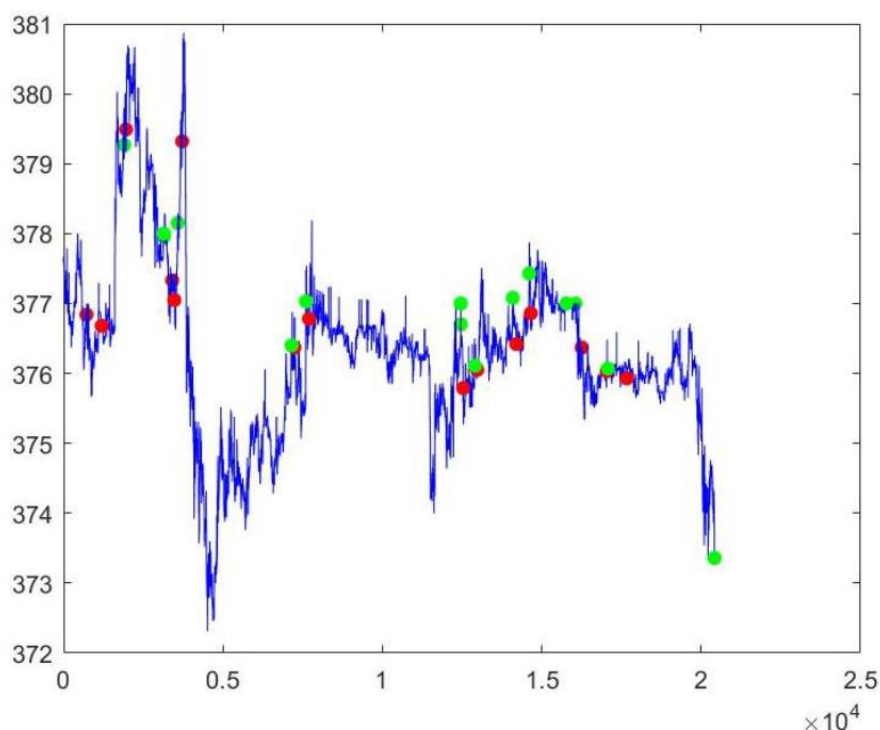


Рисунок 1.5 – Результат виконання методу прогнозування [31]

## Висновки до розділу 1

В розділі 1 проведено аналіз принципів функціонування та ціноутворення криптовалюти Біткоїн.

Актуальність дослідження полягає в інноваційності мережі платежів та цифрової валюти Біткоїн. Наразі існує потреба в інструментах, які дозволяють аналізувати великі об'єми даних, що генерують біржі обміну криптовалют – Binance, EXMO, Bittrex, щоб отримувати корисну інформацію, придатну для комерційного застосування, такого, як можливість прогнозування його коливань.

В розділі було доведено, що блокчейн стрімко зростає та розвивається в безлічі різноманітних галузей [32], що прагнуть скорочення витрат, підвищення ефективності діяльності і відмови від застарілих інформаційних структур. Приклади сфер, де вже на практиці розпочинається втілення інноваційних ідей блокчейна [33]: управління ідентифікаційною інформацією, захист авторського права, цифрові активи і токенизація, смарт-контракти, міжнародні платежі [34], держава в смартфоні [35]. Основні переваги криптовалюти: нульові або дуже низькі комісії, миттєві перекази, безпека, надійність, захищеність, можливість відправляти мікротранзакції, наднаціональна децентралізована система, відкрита інноваційна технологія з високим рівнем безпеки транзакцій, розширення задоволення зростаючих потреб у фінансових й інвестиційних послугах, ринкове ціноутворення, позбавлення від фінансових посередників, таких як фінансові установи, повний контроль над власними засобами, недоторканність приватного життя, усунення непрозорості.

Отже був обраний метод використання мереж Байєса для прогнозування курсу криптовалюти Біткоїн, так як цей інструмент дозволить провести ефективний аналіз численних незалежних функцій, що пов'язані з мережею Біткоїн та ринком і торгівлею на ньому. Ці функції були обрані через їх тісний зв'язок з ціноутворенням Біткоїн, і включають такі поняття, як ринкова

капіталізація Біткоїн, а також відношення обсягу трансакцій Біткоїн до обсягу USD.

## РОЗДІЛ 2 МЕРЕЖІ БАЙЄСА

### 2.1 Мережі Байєса як інструмент для прогнозування курсу Біткоїн

Мережі Байєса є ефективним інструментом для аналізу великих об'ємів даних. Тому, метою написання даного розділу є аналіз мереж Байєса для виокремлення тих елементів теорії цих мереж, які будуть корисними для такого прогнозування.

Для цього спочатку необхідно зрозуміти, на основі чого необхідно будувати прогноз. Користуючись розробками Isaac Madan, Shaurya Saluja та AoJia Zhao [36], було виявлено, що ефективно прогнозувати курс Біткоїн можна використовуючи незалежні функції, що пов'язані з мережею Біткоїн та ринком і торгівлею на ньому. Ці функції були обрані через їх тісний зв'язок з ціноутворенням Біткоїн, і включають такі поняття, як ринкова капіталізація Біткоїн, а також відношення обсягу трансакцій Біткоїн до обсягу USD. Дані складаються зі спостережень, що збираються щодня протягом останніх 10 років, з моменту появи Біткоїн. Інформація була отримана за допомогою ресурсу <https://www.quandl.com> [37]. 24-х годинний крок часового ряду мінімізує проблеми з шумом від вимірювань більш високої зернистості та хвилинної волатильності, а також служить для визначення конкретних аспектів, які були релевантними в прогнозуванні ціни криптовалюти Біткоїн.

У методі передбачено розглядати різницю кожної з цих змінних, щоб передбачити ознаку зміни ціни, на відміну від самої фактичної ціни, щоб представити цю проблему біноміально, для застосування до прогнозу мережі Байєса. Для використання дискретної мережі Байєса також проводитиметься попередня обробка даних – дискретизація за принципом однакової кількості елементів.

Вхідний набір даних розбиватиметься у відношенні 70:30 на навчальні дані та тестові дані. На 70% початкового набору даних проводитиметься побудова структури мережі Байєса та навчання параметрів мережі Байєса. Використовуючи алгоритм знаходження точного ймовірнісного висновку



будуватиметься прогноз для цільової вершини 30% початкового набору даних та буде сформована нова колонка зі спрогнозованими значеннями вузла. Процес побудови прогнозу для кожної точки перевірного набору даних зображений на рис. 2.1:

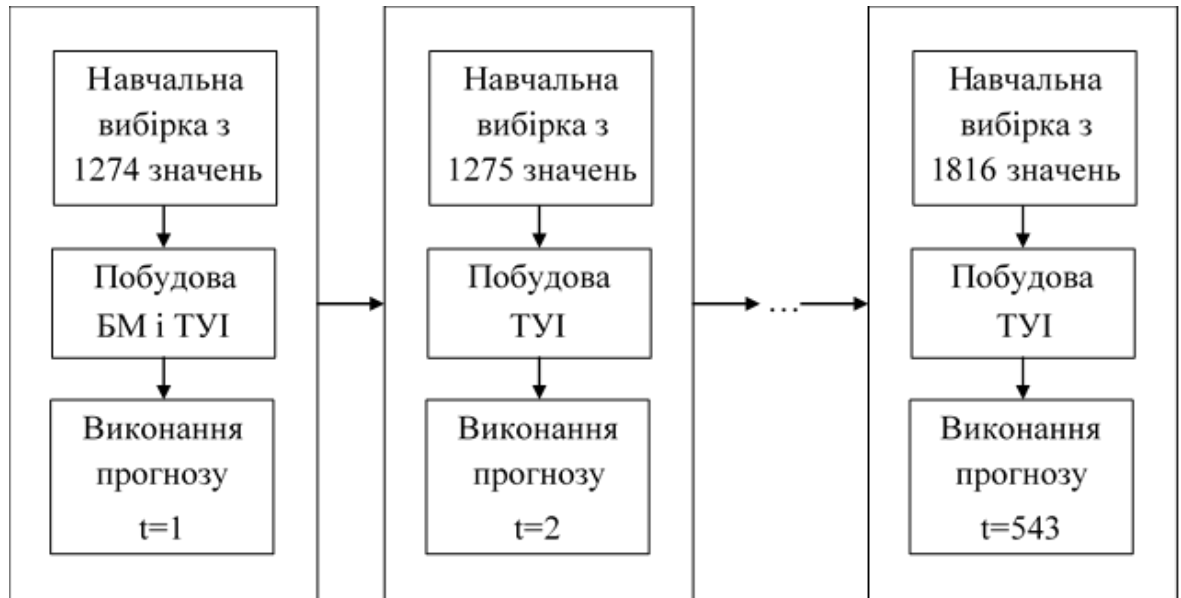


Рисунок 2.1 – Структурна схема послідовності прогнозування криптовалюти Біткоїн мережею Байєса [38]

Таким чином у результаті двійкової класифікації утвориться біноміальна змінна що відображає прогноз зміни курсу криптовалюти Біткоїн на наступну добу – зростання або спад. Вона забезпечуватиме роботу побудованої СППР з прогнозування курсу криптовалюти Біткоїн та формування інвестиційної стратегії задля зниження відповідних ризиків.

Отже, для досягнення поставленої мети необхідно більш детально дослідити мережі Байєса, їх типи, методи побудови структури таких мереж за навчальними даними, методи знаходження в них точного ймовірного висновку, та обрати ті з них, що будуть використані у сформульованому методі прогнозування курсу Біткоїн.

В таблиці 2.1 наведено розташування змінних TP, TN, FP, FN в матриці помилок, що використовується в магістерській дисертації для порівняння якості прогнозування курсу Біткоїна з аналогічними результатами інших робіт у цій сфері. Вибір таких параметрів для оцінки якості побудованого прогнозу

зумовлений роботою авторів Isaac Madan, Shaurya Saluja та Aojia Zhao [29], присвяченою цій темі.

Таблиця 2.1 – Матриця помилок (Confusion matrix)

		Спрогнозовані	
		-1	1
Дійсні	-1	223 True positives	3 False positives
	1	49 False negatives	267 True negatives

Sensitivity (TPR) визначається за формулою (2.1) [39]:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \quad (2.1)$$

Specificity (TNR) визначається за формулою (2.2) [39]:

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} \quad (2.2)$$

Precision (PPV) визначається за формулою (2.3) [39]:

$$PPV = \frac{TP}{TP + FP} \quad (2.3)$$

Accuracy (ACC) визначається за формулою (2.4) [39]:

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

## 2.2 Дослідження типів мереж Байєса

Мережа Байєса [40] – це пара  $\langle G, B \rangle$ , де перша компонента  $G$  - це спрямований ациклічний граф, що символізує випадкові змінні і формалізується як набір умов незалежності: змінні не залежать від власних батьків в  $G$ . Інша компонента пари мережі Байєса  $B$  - це множина параметрів, які утворюють мережу. Компоненти мережі містять наступні параметри [40]

$$\theta_{x^{(i)}|pa(x^{(i)})} = P(X^{(i)}|pa(X^{(i)}))$$

щодо кожного значення з обсягу можливих значень змінної  $x^{(i)} \in X^{(i)}$  та змінної  $pa(X^{(i)}) \in Pa(X^{(i)})$ , де  $Pa(X^{(i)})$  визначає  $X^{(i)} \in G$  - множину батьків змінної [40].

Довільна змінна  $X^{(i)} \in G$  представляється в вигляді вершини. При умові, якщо розглядають більше одного графа, для встановлення батьків змінної  $X^{(i)}$  у графі  $G$  використовують позначення  $P_a^G(X^{(i)})$ . Повна спільна імовірність МБ може бути обчислена за формулою (2.5) [41] наступним чином:

$$P_B(X^{(1)}, \dots, X^{(N)}) = \prod_{i=1}^N P_B(X^{(i)}|Pa(X^{(i)})) \quad (2.5)$$

З точки зору математики, мережа Байєса є моделлю представлення існуючих та відсутніх імовірносних залежностей. Причинним зв'язок  $A \rightarrow B$  стає у тому випадку, коли подія  $A$  це є причина виникнення події  $B$  у випадку, якщо існує певний механізм, за умови реалізації якого, значення, прийняте подією  $A$ , має подальший вплив на значення, прийняте подією  $B$  [42].

Мережа Байєса називається причинною (по-іншому, каузальною), якщо всі зв'язки в компонентах мережі є зв'язками причинними.

Такі три типи зв'язків в мережі існують між вершинами мережі:

- а) лінійний
- б) розбіжний (divergent)
- в) збіжний (convergent)

Приклади зв'язків ілюстровані на рис. 2.2 в трьох виглядах

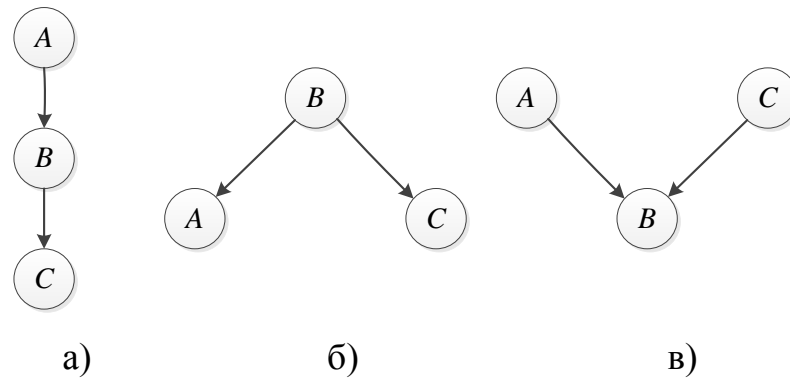


Рисунок 2.2 – Типи зв'язків [41]

Такі типи мереж Байєса існують:

1. дискретний
2. гібридний
3. неперервний
4. динамічний [41].

Дискретні мережі Байєса – мережі з дискретними змінними вершин, це означає, що вони мають скінченну кількість станів [43].

Для характеризування імовірнісного розподілу дискретної випадкової величини звичайно використовують таблицю – ряд розподілу. В мережі Байєса такий ряд записується у вигляді таблиці умовних імовірностей.

Рис. 2.3 демонструє приклад дискретної мережі для діагностування причини несправності електричної бритви, де вершина *A* діагностована як збій напруги; вершина *B* діагностована як ушкодження елементів електроніки; вершина *C* діагностована як ушкодження ріжучого леза; вершина *D* діагностована як несправність електричної бритви.

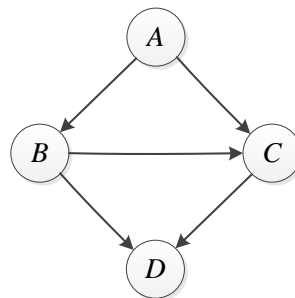


Рисунок 2.3 – Приклад дискретної Мережі Байєса [41]

Мережі відповідає певний набір таблиць деяких умовних імовірностей вершин (табл. 2.2–2.5), у якій 1 - позитивна відповідь, 0 – негативна, якщо, наприклад,  $B=0$ , то це означатиме, що електроніка неущокоджена:

Таблиця 2.2 – Значення умовної імовірності А

Стан вершини	Значення імовірності
$A=0$	0,8
$A=1$	0,2

Таблиця 2.3 – Умовна імовірність В

Батьки вершини	Стани батьків вершини	
$A$	$A=0$	$A=1$
Стан вершини	Значення імовірності	
$B=0$	0,8	0,2
$B=1$	0,2	0,8

Таблиця 2.4 – Умовна імовірність С

Батьки вершини	Стани батьків вершини			
$A$	$A=0$	$A=0$	$A=1$	$A=1$
$B$	$B=0$	$B=1$	$B=0$	$B=1$
Стан вершини	Значення імовірності			
$C=0$	0,85	0,5	1	0,15
$C=1$	0,15	0,5	0	0,85

Таблиця 2.5 – Умовна імовірність D

Батьки вершини	Стани батьків вершини			
$B$	$B=0$	$B=0$	$B=1$	$B=1$
$C$	$C=0$	$C=1$	$C=0$	$C=1$
Стан вершини	Значення імовірності			
$D=0$	1	0	0,5	0
$D=1$	0	1	0,5	1

Неперервні Мережі Байєса – це мережі, в яких змінні вершин неперервні.

Існує багато випадків, де події можуть мати будь-який стан з певного допустимого діапазона. Змінна  $X$  стане неперервною випадковою величиною, отже множиною можливих станів будуть всі значення, що вона становить:  $X = \{x : a \leq x \leq b\}$ .

Щодо імовірносного розподілу для неперервної випадкової величини можна використати функцію розподілу або щільність розподілу.

До прикладу, вершина  $X$  має певну множину вершин – її батьків –  $A = \{A_1, \dots, A_n\}$ , то умовний розподіл для випадкової величини записується за допомогою формули (2.6) [44]:

$$f(X / A_i) = N(\mu_x + k_i \cdot \mu_i; \sigma_x), \quad (2.6)$$

де  $\mu_x$  – математичне сподівання для величини  $X$ ;  $\mu_i$  – математичне сподівання для величини  $A_i$ ;  $k_i$  – ваговим коефіцієнт, який відображає зв'язок між самою вершиною та її вершиною-предком  $A_i$ ;  $\sigma_x$  – стандартне відхилення для величини вершини  $X$ . Тоді розподіл обчислюється за формулою (2.7) [44]:

$$N(\mu_x + k_i \cdot \mu_i; \sigma_x) = \frac{1}{\sigma_x \sqrt{2\pi}} \cdot \exp \left[ -\frac{1}{2} \left( \frac{x - (\mu_x + k_i \cdot \mu_i)}{\sigma_x} \right)^2 \right] \quad (2.7)$$

Лінійною регресійною моделлю можна представити зв'язок між змінною  $X$  та її вершинами-предками:  $X = k_1 \cdot A_1 + \dots + k_n \cdot A_n + Q_x$ , де  $k_i$  – регресійним коефіцієнтом, що представляє зв'язок зпоміж змінними  $X$  та  $A_i$ ;  $Q_x$  – так звана шумова компонента, що може бути представлена у вигляді нормального розподілу із нульовим математичним очікуванням.

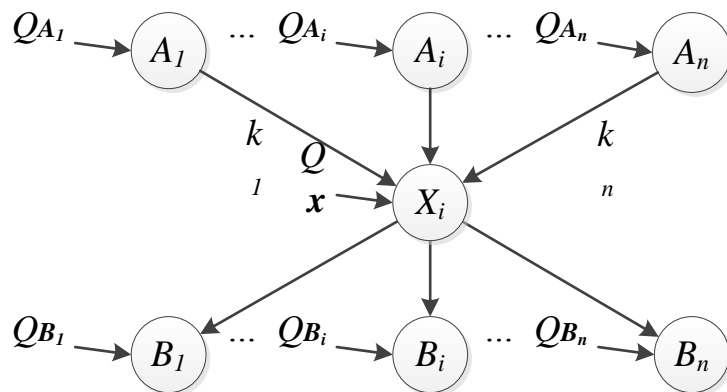


Рисунок 2.4 – Приклад неперервної Мережі Байєса [44]

Рис. 2.4 наочно демонструє неперервну МБ, що відповідає нормальному закону розподілу, в якому відповідно  $A_1, \dots, A_n$  є предками  $X$ ;  $B_1, \dots, B_n$  – є відповідно її нащадками;  $k_1, \dots, k_n$  являються ваговими коефіцієнтами; відповідно  $Q_{A1}, \dots, Q_{An}, Q_x, Q_{B1}, \dots, Q_{Bn}$  – є шумовими коефіцієнтами.

Гібридними Мережами Байєса називаються мережі, вершини в яких містять як дискретні змінні так і неперервні.

Одночасна наявність дискретних й також неперервних змінних в одній й тій же самій мережі накладає суттєві обмеження:

- дискретні вершини в жодному разі не можуть бути нащадками неперервних предків,
- неперервні змінні задовольняють певним критеріям нормального закону розподілу, і мають умовний з приводу значень їх предків,
- нормальним являється розподіл неперервної змінної  $X$  із дискретними вершинами – предками  $A$  і неперервними предками – вершинами  $B$ , тоді ймовірність можна обчислити за формулою (2.8) [44]:

$$P(X / A = a, B = b) = N(\mu_x(\mu_A, \mu_B); \sigma_x(\sigma_A)), \quad (2.8)$$

де  $\mu_x, \mu_A, \mu_B$  являються математичними очікуваннями,  $\sigma_x, \sigma_A$  являються стандартними відхиленнями.  $\mu_x$  лінійно залежить від неперервних предків,  $\sigma_x$  від них не залежна [44]. Однак від дискретних предків залежні  $\mu_x$  і  $\sigma_x$ .

Припустимо, що дискретні випадкові величини  $A_1, \dots, A_n$  разом з неперервними випадковими величинами  $B_1, \dots, B_m$  впливають певним чином на результуючу випадкову величину  $X$ , що є неперервною, як проілюстровано згідно рис. 2.5. Вершини із квадратними рамочками мають дискретний розподіл, вершини із круглими рамочками мають розподіл неперервний.

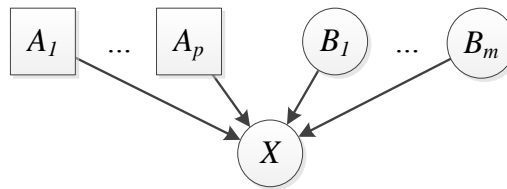


Рисунок 2.5 – Приклад гібридної Мережі Байєса [44]

Кожна з дискретних випадкових величин, що розглядається в даному розділі,  $A_j$  ( $j = 1, \dots, p$ ) може приймати певні значення  $A_{ij}$  ( $i = 1, \dots, n_j$ ) з такими імовірностями  $P_{ij}$ , для яких  $\sum_{i=1}^{n_j} P_{ij} = 1$ . Загалом вплив дискретних випадкових величин на величину  $X$  описується математичним сподіванням ( $\mu_{i1}, \dots, \mu_{ip}$ ) та стандартним відхиленням ( $\sigma_{i1}, \dots, \sigma_{ip}$ ). Кожна випадкова величина із переліку неперервних випадкових величин  $B$  ( $l = 1, \dots, m$ ) розподілена за нормальним розподілом, його параметри ( $\mu_l; \sigma_l$ ). Загалом вплив оцих неперервних величин разом із значеннями дискретних величин на кінцеву випадкову величину  $X$  характеризується наступними ваговими коефіцієнтами  $k_l, i1, \dots, ip$  для усіх  $l = 1, \dots, m$ .

Тому характеристики величини  $X$  можна обчислити за формулами (2.9) та (2.10) [44]:

$$\mu_x = \sum_{i1=1}^{n_1} \dots \sum_{ip=1}^{n_p} P_{1,i1} \dots P_{p,ip} \cdot \left( \mu_{i1, \dots, ip} + \sum_{l=1}^m k_{l,i1, \dots, ip} \cdot \mu_l \right) \quad (2.9)$$

$$\sigma_x = \sqrt{\sum_{i1=1}^{n_1} \dots \sum_{ip=1}^{n_p} P_{1,i1} \dots P_{p,ip} \cdot \left( \left( \mu_{i1, \dots, ip} + \sum_{l=1}^m k_{l,i1, \dots, ip} \cdot \mu_l \right)^2 + \sigma_{i1, \dots, ip}^2 + \sum_{l=1}^m k_{l,i1, \dots, ip}^2 \cdot \sigma_l^2 \right) - \mu^2} \quad (2.10)$$

Динамічні мережі Байєса використовують для моделювання часових процесів, вони є мережами, в яких значення вузлів із часом змінюється. Ще динамічні мережі Байєса (dynamic Bayesian network) можуть мати назву «часової» мережі Байєса (temporal Bayesian network), й цей термін є навіть більш точним тому, що, хоча для опису поточного стану процесу можна



додавати також ще й приховані вузли, все ж структура моделі залишається незмінною [44].

В прихованій моделі Маркова (hidden Markov model) у кожному шарі мережі є одна прихована дискретна вершина, і ще по одній спостережуваний дискретній або неперервній вершині. Це найпростіший тип серед динамічних мереж Байєса. Рис. 2.6 унаочнює такий варіант моделі, вершини в прямокутних рамочках містять дискретні змінні, в овальних рамочках мають неперервні змінні.  $Y$  є спостережуваними вершинами, а  $X$  є прихованими вершинами.

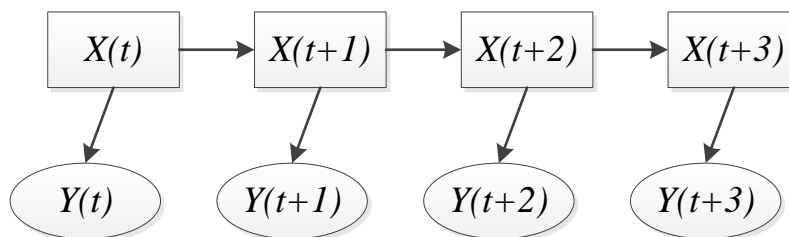


Рисунок 2.6 – Приклад динамічної Мережі Байєса [44]

Необхідність визначення  $P(X(t))$  - початкового розподілу, топології у шарі  $P(X(t+i) | X(t+i-1))$  та поміж шарами –  $P(Y(t) | X(t))$ , з'являється задля визначення динамічної мережі.

Розпізнавання мовлення є складним прикладом, який ілюструє аспекти використання динамічних мереж Байєса. Там  $Y$  є фонемами - спостережувані змінні, а  $X$  – літери з вимовлених слів є прихованими. Така мережа Байєса є множиною багаторазово повторюваних блоків в часовому інтервалі [45].

Підсумовуючи розглянуті типи мереж, можна прийти до висновку, що необхідно використати дискретні мережі Байєса для прогнозування курсу криптовалюти Біткоїн, через дискретну природу досліджуваних навчальних даних.

### 2.3 Побудова мереж Байєса

Вираз «побудувати мережу Байєса» означає вибрати оптимальну структуру мережі. Саме оптимальність визначає, що обрана структура найбільш правдоподібно відповідатиме наявним в розпорядженні навчальним

даним або модельованому процесу. В основі кожного методу побудови мереж Байєса закладена формула Байєса [46].

Введемо наступні позначення. Літерою  $\Omega$  будемо позначати вибіркоий простір чи простір подій (тобто сукупність всіх елементарних подій). Зробимо припущення: нехай є дві події  $A \in \Omega$  і  $B \in \Omega$ .

Імовірність такого збігу обставин, що подія  $A$  відбудеться за умови, щоб відбулась подія  $B$ , позначимо  $P(A / B)$  та назвемо умовною імовірністю події  $A$  при настанні заданої події  $B$ . Імовірність ситуації, коли обидві події таки відбудуться  $P(A \cap B)$  будемо називати сукупною імовірністю.

Умовна імовірність  $P(A / B)$  для подій  $A, B$  визначається відношенням сукупної імовірності до ймовірності  $B$ , за умови, що не рівняється нулю, за формулою Байєса (2.11) [46]:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (2.11)$$

Відповідно формула (2.12) [46] для  $B$  :

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \quad (2.12)$$

Через властивість комутативності для сукупної імовірності  $P(A \cap B) = P(B \cap A)$ , отримуємо правило Байєса (2.13) [46]:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (2.13)$$

Якщо не перетинаються одна з іншою множини  $A$  і  $B$  , тоді маємо формулу (2.14) [46]:

$$P(A \cup B) = P(A) + P(B) \quad (2.14)$$

Згідно із відомою теорією множин  $B$  записується формулою (2.15) [46]:

$$B = (B \cap A) \cup (B \cap \neg A) \quad (2.15)$$

Ці об'єднання не перетинаються, відповідно маємо формулу (2.16) [46]

$$\begin{aligned} P(B) &= P((B \cap A) \cup (B \cap \neg A)) = P(B \cap A) + P(B \cap \neg A) = \\ &= P(B | A) \cdot P(A) + P(B | \neg A) \cdot P(\neg A) \end{aligned} \quad (2.16)$$

Якщо перейти від розгляду множин до розгляду подій, отримаємо можливість підставити вираз в формулу Байєса, формула (2.17) [46]:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B|A) \cdot P(A) + P(B|\neg A) \cdot P(\neg A)} \quad (2.17)$$

В термінах гіпотез  $H$  та спостережень  $E$  формула (2.18) [46] пишеться так:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E|H) \cdot P(H) + P(E|\neg H) \cdot P(\neg H)} \quad (2.18)$$

Цей вираз є наочною ілюстрацією до зв'язку спостереження із поки ще не підтвердженою гіпотезою, та гіпотези зі спостереженням. Зазначається, що імовірність  $P(E | H)$  та апіорна імовірність гіпотези  $P(H)$  визначені ще до отримання спостережень. Тому для імовірності  $P(H | E)$  використовують термін апостеріорної імовірності гіпотези  $H$  по спостереженню  $E$ .

Для отримання формули (2.19) [46] узагальнюють для випадку  $n$  гіпотез,:

$$P(H_k|E) = \frac{P(E|H_k) \cdot P(H_k)}{\sum_{i=1}^n P(E|H_i) \cdot P(H_i)} \quad (2.19)$$

Ця узагальнена формула є формула Байєса. На ній будують мережі Байєса. Величина  $H_k$  визначає гіпотезу серед кількості  $n$  можливих. Імовірності  $P(E|H_k)$  або апріорно задають експерти, або виконують розрахунки за навчальними даними, що є в розпорядженні.

Вислів «побудова мережі Байєса» має на увазі розв’язування поставлених завдань:

1. пошук і визначення оптимальної структури мережі Байєса, такого направленного ациклічного графу, який максимально чітко відповідає навчальним даним або тому процесу, що піддається аналізу;
2. обчислення для кожної з вершин цього графа відповідних значень таблиць умовних імовірностей мережі Байєса.

Перешкодою існуючим методам стають наступні проблеми та виклики:

1. Наявна впорядкована множина вершин. Для переважної кількості методів, особливо старих, множина вважається визначеною, але реально в житті це почасти не відповідає дійсності, особливо при роботі із фактичними даними.
2. Низький ступінь обчислювальної ефективності. Наявні методи функціонують без використання впорядкованої множини вершин, замість цього використовується тестування умовної незалежності. Існує потреба у виконанні експоненціальної кількості тестувань, це призведе до зменшення ефективності застосування даного методу завдячуючи зростанню обсягу потрібних обчислень.
3. Значним викликом є можливість практичної реалізації мережі Байєса зі значною кількістю вершин. Наразі розроблені декілька методів, здатних до побудови структури мереж Байєса навіть із кількома сотнями вершин, та використовуючи навчальну вибірку, налічуючу понад мільйони записів. До прикладу, це такі методи, як Tetrad II [47] й SopLeq [48].

Для описування мереж Байєса визначають топологію графа (тобто будову його структури) і параметри щодо кожної з вершин. Таку інформацію на практиці можна отримати з наявних навчальних даних, на відміну від отримання вірної топології мережі, що є більш складним викликом порівняно з завданням визначення параметрів вершин. Окремий власний підхід потрібний в ситуації, коли наявне приховування якихось вершин або ж некоректних чи можливо неповних даних. Дослідники для вирішення таких типів задач виокремлюють різні варіанти навчання мереж Байєса. До прикладу, коли структура мережі вже є відомою, і також наявний повний обсяг спостережень, в цьому випадку доречно буде використати метод максимальної оцінки правдоподібності. Якщо структура є невідома, виконується пошук в просторі моделей. Ситуація для аналітика може скластися й таким чином, що аналітик володіє частковими спостереженнями, в таких обставинах доцільно застосувати градієнтний метод, або ЕМ-алгоритм, чи обрати рішення задачі за допомогою вибірки Гіббса. В складній ситуації, коли невідома структура, доцільно використати чи структурний ЕМ-алгоритм, чи алгоритм стиснення кордонів[49].

### 2.3.1 Випадок відомої структури та повних спостережень

За таких обставин обчислюються значення параметрів щодо кожного з умовних імовірнісних розподілів, це максимізує правдоподібність наявних навчальних даних, що містять  $M$  записів (припускаючи, що ці записи є незалежні). Нормалізоване логарифмічне рівняння щодо правдоподібності навчальної множини  $D = \{D_1, \dots, D_M\}$  може бути записане формулою (2.20) [41]:

$$L = \frac{1}{M} \sum_{i=1}^n \sum_{m=1}^M \log P(X_i | Pa(X_i), D_m), \quad (2.20)$$

де  $Pa(X_i)$  – множина предків вершини  $X_i$ ,

$D$  – наявна навчальна вибірка,

$D_m$  – це є елемент із  $m$ -ї наявної навчальної вибірки,

$M$  – кількість подій у цій навчальній вибірці.

До прикладу, розглянемо певну просту мережу, що описує процес роботи сервісного центру по діагностиці ПК. Схема мережі зображається на рис. 2.7. Кожна її вершина може перебувати в 2-ох можливих станах: «присутності» і «відсутності» для вершин А – це збій в роботі модуля ПК, В – невірна робота плати пам'яті, С – неможливість стартувати для ОС при початковому запуску ПК, D – відправка на гарантійний або поточний ремонт в сервісний центр, Е – переустановка операційної системи.

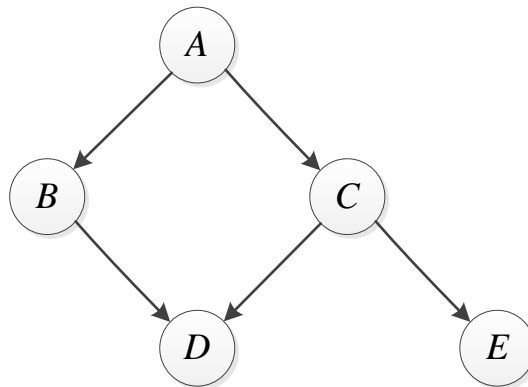


Рисунок 2.7 – Байєсівська мережа випадку «Сервісний центр» [41]

Для реалізованого дискретного випадку оцінка вузла D обчислюється таким способом. Припустімо, наявні дані для навчання, тобто відома інформація щодо кількості випадків  $N(D=1, C=0, B=1)$ ,  $N(D=1, C=1, B=0)$  й так далі й далі. Використовуючи цю інформацію, максимальна оцінка правдоподібності щодо вершини D обчислюється по формулі (2.21) [41]:

$$\begin{aligned}
 P(D = d | B = b, C = c) &= \frac{N(D = d, C = c, B = b)}{N(C = c, B = b)} = \\
 &= \frac{N(D = d, C = c, B = b)}{N(D = 0, C = c, B = b) + N(D = 1, C = c, B = b)}
 \end{aligned}
 \tag{2.21}$$

Якщо вершини описані функцією Гауса, в цих обставинах можна підрахувати вибіркове середнє і дисперсію, далі за допомогою лінійної регресії можна обчислити матрицю ваг [41]. Інші види розподілів передбачають необхідність використання складніших процедур.

### 2.3.2 Випадок невідомої структури й повних спостережень

Найбільш імовірною моделлю в цих обставинах є повний граф, в такому випадку є задіяною найбільша кількість параметрів, і тому така модель найбільш відповідатиме наявним даним.

Формула Байєса (2.22) [41] тоді виглядатиме так:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)}, \quad (2.22)$$

тут  $G$  – направлений ациклічний граф, що відповідає випадковим змінним,

$D = \{x^1, \dots, x^N\}$  – множина даних.

Прологарифмуючи формулу, отримаємо формулу (2.23) [41]:

$$\log(P(G|D)) = \log(P(D|G)) + \log(P(G)) + (-\log(P(D))) \quad (2.23)$$

Тут у формулі один з доданків, а саме,  $\log(P(D))$  відіграє роль такої собі штрафуючої компоненти за надскладні моделі. У формулі її складова  $P(D|G)$  також може стати штрафуючою компонентою за ситуацію надто складної моделі.

Для завершення виконання точних розрахунків, які пов'язані з вибором моделі треба вираховувати  $P(D) = \sum_G P(D|G)$ , це завдання має експоненційну складність.

Взамін цього можна скористатися Байєсовим інформаційним критерієм, що будується за формулою (2.24) [41]:

$$\log(P(G|D)) \approx \log(P(D|G, \hat{\theta}_G)) - \frac{\log(N)}{2} \dim(G), \quad (2.24)$$

тут  $N$  задає число моделей,

$\dim(G)$  – це є кількість вільних параметрів,

$\hat{\theta}_G$  – становить максимально правдоподібну оцінку параметрів,

$\dim()$  – є штрафуючою компонентою за надскладні моделі.

На етапі, наступному після вибору структури, відбувається процес навчання цієї структури, задля того, щоб направлений ациклічний граф найкраще задовольняв наявним даним. Це завдання характеризується як NP-складне завдання, тобто є задачею з нелінійною поліноміальною оцінкою числа ітерацій. Саме того, найчастіше використовуються саме локальні алгоритми пошуку, до прикладу, метод гілок і меж для пошуку та винайдення в просторі графів чи популярний метод «жадібного» знаходження екстремуму (greedy hill climbing method).

ЕМ-алгоритм (англійською мовою Expectation-maximization (EM) algorithm) [41] використовується для оцінки на максимальну схожість параметрів змовірних моделей, у випадку, коли модель залежить від прихованих змінних. Кожна з ітерацій алгоритму складається з двох етапів. На Е-кроці (expectation) обчислюється очікуване значення функції правдоподібності, і приховані змінні розцінюються як спостережувані. На М-етапі (maximization) обчислюється оцінка максимальної подібності, отже збільшується очікувана подібність, що була вирахована на Е-кроці. На наступному етапі це значення використовують для Е-кроку на наступній ітерації. Надалі алгоритм виконується до збіжності.

Часто ЕМ-алгоритм використовується для розділення суміші функції Гауса.

Розглянемо функцію, що записана формулою (2.25) [41]:

$$F(q, \theta) = E_q[\log L(\theta; x; Z)] + H(q) = \quad (2.25)$$



$$= -D_{KL}(q||p_{z|x}(\cdot|x;\theta)) + \log L(\theta;x),$$

де  $q$  — розподіл імовірностей неспостережуваних змінних  $Z$ ;

$p_{Z|X}(\cdot|x;\theta)$  — це умовний розподіл неспостережуваних змінних при фіксованих спостережуваних змінних  $x$  і параметрах розподілу імовірностей неспостережуваних вузлів  $\theta$ ;

$H$  — ентропія;

$D_{KL}$  — відстань Кульбака — Лейблера, що визначається за формулою (2.26) [41]

$$D_{KL}(P||Q) = - \sum_{x \in X} P(x) \log \left( \frac{Q(x)}{P(x)} \right) \quad (2.26)$$

Тоді етапи ЕМ-алгоритму можна записати як:

Е(xpectation) етап: Знаходимо  $q$ , щоб максимізувати  $F$ , формула (2.27) [41]:

$$q^{(t)} = \text{*argmax}_q F(q, \theta^{(t)}) \quad (2.27)$$

М(aximization) етап: Знаходимо  $\theta$ , щоб максимізувати  $F$ , формула (2.28) [41]:

$$\theta^{(t+1)} = \text{*argmax}_\theta F(q^{(t)}, \theta) \quad (2.28)$$

Розроблені наразі методи побудови структури мереж Байєса, зручно розділити на дві великі категорії: що ґрунтуються на оціночних функціях (search & scoring) та із застосуванням тесту на наявну умовну незалежність (dependency analysis).

### 2.3.3 Евристичний метод побудови Мереж Байєса

Різноманіття можливих форм мережі Байєса росте пропорційно кількості вузлів. До прикладу, наявний спосіб побудови 25 окремих ациклічних спрямованих графів, що складаються з 3 вузлів. Ця кількість зростає до 543 для 4-х змінних, 29281 для 5-ти змінних і 3781503 мереж з 6-ти вузлів [38]. За існуючих обставин, коли потрібно визначити усі зв'язки між 15-ти змінними, необхідно обрати лише тільки одну оптимальну мережу Байєса з-поміж приблизно  $2.4 \cdot 10^{41}$  можливих структур [38]. Отже, актуальною є потреба в спеціальній процедурі пошуку такої оптимальної структури без обов'язкової перевірки всіх. Евристичний метод побудови ґрунтується на взаємній інформації для пар змінних  $x^i$  та  $x^j$  у наборі даних, обчислюваній за формулою (2.29) [38]

$$MI(x^i, x^j) = \sum_{x^i, x^j} P(x^i, x^j) \cdot \log \left( \frac{P(x^i, x^j)}{P(x^i)P(x^j)} \right) \quad (2.29)$$

Це значення пропонує інформацію щодо порядку побудови оптимального набору потрібних даних. Побудова мережі завжди починається із пари вузлів, що має найбільше із значень  $MI$ . Наявні три можливих результати для кожної з таких пар: два способи для випадку, де одна змінна є батьківською для іншої змінної, і одна можливість для них – це не бути безпосередньо зв'язаними. Тож, метод вибирає з цих трьох найкращі результати, використовуючи значення іншого вимірювання, а саме, функції опису мінімальної довжини (MDL), використовуючи теорію кодування, запропоновану Шенноном [43]. Вчений [43] стверджує наступне: існує фіксована мінімальна довжина коду, що закодує певне повідомлення, яке повністю залежить від інформації в цьому повідомленні. Тож, якщо код є базованим на неправильному сприйнятті даного повідомлення, тоді його довжина буде більше за оптимальну. При застосуванні в аспекті мереж Байєса ця формула (2.30) [38] виглядатиме наступним чином:

$$L(j, g, x^n) = H(j, g, x^n) + \frac{k(j, g)}{2} \cdot \log(n) \quad (2.30)$$

для кожної зі змінних в наборі наявних даних,

тут  $n$  – це є кількість спостережень,

$g$  – це є поточна структура мережі,

$j$  – це є число вузла;

$x^n = d_1 \dots d_n$  це є набір наявних даних, в якому кожна змінна  $j$  має  $A^{(j)}$  усіляких різних станів,

$k(j, g)$  – це є число незалежних імовірностей  $j$ -змінної.

Літерою  $H$  позначається емпірична ентропія, яка обчислюється за формулою (2.31) [38]:

$$H(j, g, x^n) = \sum_{(j,g)} \sum_{q \in A^{(j)}} -n[q, s, j, g] \cdot \log\left(\frac{n[q, s, j, g]}{n[s, j, g]}\right) \quad (2.31)$$

У формулі ентропії необхідні додаткові розрахунки для обчислення значення формул (2.32) [38] та (2.33) [38]:

$$n[s, j, g] = \sum_{i=1}^n I\left(\pi_i^{(j)} = s\right) \quad (2.32)$$

$$n[q, s, j, g] = \sum_{i=1}^n I\left(X_i = q, \pi_i^{(j)} = s\right), \quad (2.33)$$

де  $\pi^{(j)} = \Pi^{(j)}$ ,  $X_k = x_k, \forall k \in \emptyset^{(j)}$ .

Функція  $I(E)=1$ , у випадку, коли предикат  $E=true$ , в іншому випадку функція  $I(E)=0$  Число незалежних імовірностей можна обчислити за формулою (2.34) [38]:

$$k(j, g) = (\alpha^{(j)} - 1) \prod_{k \in \emptyset(j)} \alpha^k, \quad (2.34)$$

де  $\emptyset(j) \subseteq \{1, \dots, j-1, j+1, \dots, N\}$  це є така множина, для якої виконується формула (2.35) [38]:

$$\Pi^{(j)} = \{X_k : k \in \emptyset(j)\} \quad (2.35)$$

$\Pi^{(j)}$  – є такою множиною батьківських вершин, де  $\Pi^{(j)} \subseteq \{X_1, \dots, X_N\} \setminus \{X_j\}$ . Розбудовуючи таким способом мережу Байєса, проводячи мінімізування MDL на кожному кроці реалізації алгоритму, й буде досягнута бажана цільова оптимальна структура. (Так ОМД мережі становить суму ОМД усіх її вершин)

Після успішного завершення процесу зчитування даних алгоритм визначає кожний зі  $\frac{n(n-1)}{2}$  унікальних пар змінних, при цьому допускаючи, що загалом в ній є наявні  $n$  вузлів, що повністю співпадає з кількістю стовпців введених даних. Метод посиляється на назви стовпчиків для ідентифікації та можливості відбору даних.

Потім за реалізацією методу повинно обчислитися значення МІ для кожної такої пари змінних. Тому на початку, першим кроком є підрахунок імовірності для кожної зі змінних в наявному наборі даних для встановлення кожного з можливих її значень.

На завершення всіх необхідних допоміжних обчислень, значення МІ оцінюються щодо кожної пари зі змінних у наборі наявних даних, використовуючи формулу.

Далі реалізується ітеративний процес по збору всіх значень  $MI$  з тимчасових наборів наявних даних в один цільовий набір, щоб отримати можливість працювати з даними в майбутньому. Далі набір відсортовується у порядку зменшення, згідно методології.

Щоб розпочати будувати усіляки різні структури мереж та оцінити їх MDL, необхідні й допоміжні обрахунки. Спершу обчислюється ентропія щодо кожної пари змінних, використовуючи для цього попередньо створені набори даних зі взаємними імовірностями.

Відібрані дані дозволяють здійснити пошук оптимальної структури. Запропонований метод створює нову матрицю, де стовпчики представляють собою пари змінних, рядочки відповідають кожній структурі в побудованій байєсівській мережі. Із першого стовпчика, рядочок збільшується втричі, задля того, щоб представити кожний можливий результат щодо залежності для певної пари вузлів. Одночасно проходить процес перевірки ациклічності отриманих графів.

Надалі значення MDL обчислюється щодо кожного рядочка певного набору даних, і лише рядочки з мінімальною можливістю MDL будуть залишені в результаті впровадження методу. Процедура продовжуватиметься для всіх  $\frac{n(n-1)}{2}$  пар до тих пір, доки ми не отримаємо оптимальну структуру мережі Байєса.

Володіючи матрицею з'єднань, отримуємо можливість побудувати й саму мережу. Отож, в результаті маємо побудовану оптимальну структуру.

Даний алгоритм побудови структури мережі Байєса за навчальними даними ідеально підходить для нашого методу прогнозування курсу Біткоїн.

## 2.4 Алгоритми імовірнісного висновку

Мережу Байєса можна визначити як імовірнісну експертну систему, в якій база знань представляється як топологія мережі й таблиця з умовних імовірностей для кожної вершини. Використання бази знань для побудови висновків, для знаходження відповідей на запити користувачів є головною філософією її будови.

В мережах Байєса закладені два головні типи знаходження висновків: імовірнісний висновок (probabilistic inference або belief updating) та максимальне апостеріорне тлумачення (belief revision, також Maximum a Posteriori – MAP explanation) [50].

Задачею імовірнісного висновку є винайдення  $P(X \mid E)$ , тобто апостеріорної імовірності вершин  $X$ , що є метою для пошуку, при певному значенні вершин  $E$ , що спостерігаються. Один із найелементарніших випадків імовірнісного висновку такий, в якому лише одна вершина для пошуку.

Метою другого типу знаходження висновків, а саме, belief revision, є виконання пошуку з ціллю знайти найбільш імовірних інстанціювань вершин, при заданому значенні спостережуваних вершин. В результаті буде отриманий певний набір інстанціювань, що може значно змінюватися, знаходячись під впливом виконуваних нових спостережень [50]. Якщо обставини складуться таким чином, що для жодної змінної, значення якої вимагається відшукати, немає спостереження, мова ведеться про найбільш імовірне тлумачення (most probable explanation – MPE). Значить, відшукування MPE символізує знаходження пояснення, де б всі інші пояснення мали б менші значення імовірностей [51].

Завдання будови імовірнісного висновку є надскладним для обчислювання і зовсім не однозначним, неможливо просто створити лише один алгоритм побудови, який би спромігся дати найліпші результати для всіх типів мереж. Виокремлені два класи імовірнісного висновку виходять з розміру поставлених завдань: чи точний чи апроксимаційний. Для реальних життєвих обставин застосування точного імовірнісного висновку не представляється можливим через надвелику складність при обчисленнях, в даних обставинах вихід у застосуванні апроксимаційних методів із обчисленням з якоюсь мірою наближення.

Необхідно розглянути існуючі алгоритми пошуку точного імовірнісного висновку у мережах Байєса та визначити той, що підходить для вирішення проблеми, що досліджується цією роботою.

### 2.4.1 Алгоритм розповсюдження повідомлення для однозв'язних мереж (Перла)

Для обрахунків значення імовірностей щодо повідомлень між вершинами – сусідками в 1986 році американським дослідником та науковцем Джуді Перлом був віднайдений [52] алгоритм обміну повідомленнями між вершинами-предками та вершинами-нащадками. В 1988 Джуді Перл [50] запропонував власну ідею реалізації алгоритму щодо розповсюдження повідомлення для однозв'язних мереж.

Однозв'язна мережа, чи полі дерево, визначається через направлений ациклічний граф, у якому існує тільки одна ненаправлена послідовність зв'язків в графі між будь-якими двома його вершинами. На рис. 2.8 можна побачити приклад такого полідерева, на рис. 2.9 – структури, які не відповідають критеріям такого визначення.

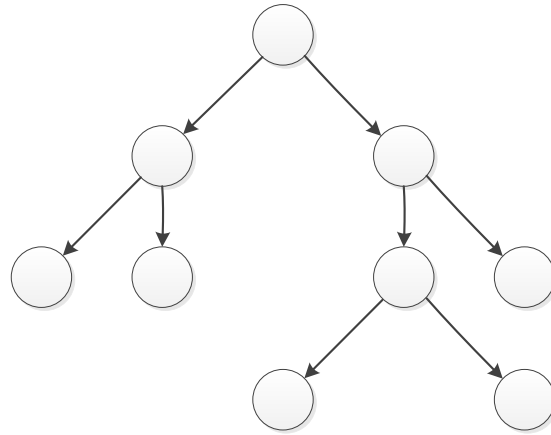


Рисунок 2.8 – Зображення полідерева [50]

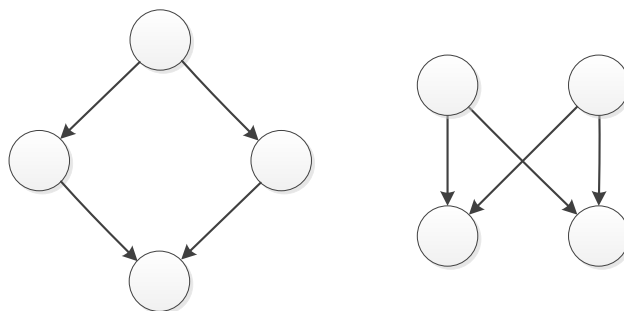


Рисунок 2.9 – Не є полідеревом [50]

Алгоритм розповсюдження повідомлення Джуді Перла [52] для однозв'язних мереж використовує цікаву ідею: вплив кожної нової порції спостережень має розглядатися наче збурення. За допомогою пересилання повідомлень між вершинами-сусідками процес збурення поширюється мережею.

Інформація щодо стану вершини-предка передається з допомогою  $\pi$ -повідомлення, а щодо вершини-нащадка – із допомогою  $\lambda$ -повідомлення – рис. 2.10.



Рисунок 2.10 – зображення  $\pi$ -повідомлення та  $\lambda$ -повідомлення [50]

На рис. 2.11 наочно продемонстрований процес розповсюдження повідомлень для наявного полідерева.



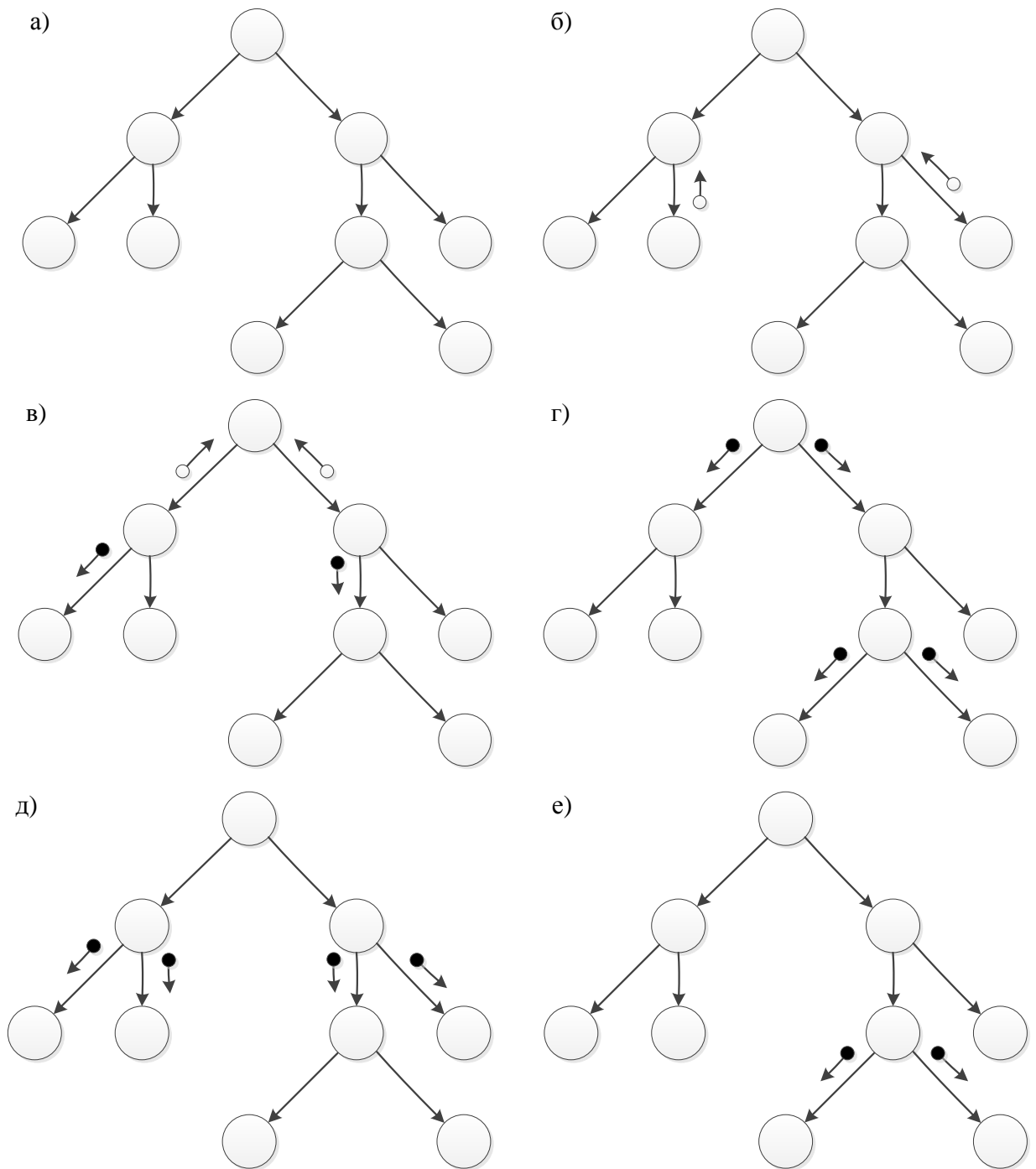


Рисунок 2.11 – Зображення процесу розповсюдження повідомлень [50]

У залежності від кількості вершин змінюється складність алгоритму.

На рис. 2.11 а) можна роздивитися схематичну структуру полідерева, де будуть розповсюджуватися повідомлення.

Рис. 2.11 б) дозволяє роздивитися як проходять інстанціювання двох змінних і розпочинають розповсюджуватись  $\lambda$ -повідомлення до самого коріння дерева.

На рис. 2.11 в) стрілочками продемонстровано принцип руху  $\lambda$ -повідомлення напрямку кореневої вершини, а от дві вершини, що вже пройшли наступний етап і отримали повідомлення на попередньому кроці, надсилають ці  $\pi$ -повідомлення своїм вершинам-нащадкам.

На рис. 2.11 г) стрілочки символізують, що кореневі вершини дерева сприйняли  $\lambda$ -повідомлення й відсилають  $\pi$ -повідомлення вже у зворотньому напрямі, тобто в напрямі своїх нащадків.

На рис. 2.11 д) стрілочки відображають подальший хід процесу розповсюдження  $\pi$ -повідомлень.

Рис. 2.11 е) ілюструє напрямок процесу, де два останні  $\pi$ -повідомлення надіслані листкам дерева.

#### 2.4.2 Алгоритм визначеного перетину

В 1988 році для дослідження розповсюдження повідомлень у багатозв'язних мережах Джуді Перлом був запропонований іще алгоритм визначеного перетину [50].

Основна концепція алгоритму полягає в зміні структури багатозв'язної мережі кількома однозв'язаними за допомогою входящих до перетину інстанціювання вершин.

На рис. 2.12 зображена реалізація схеми алгоритму на прикладі розподілення багатозв'язної мережі на дві однозв'язні мережі. Такий перетин містить одну вершину мережі, яка приймає один з двох можливих станів  $\{A_1, A_2\}$ .

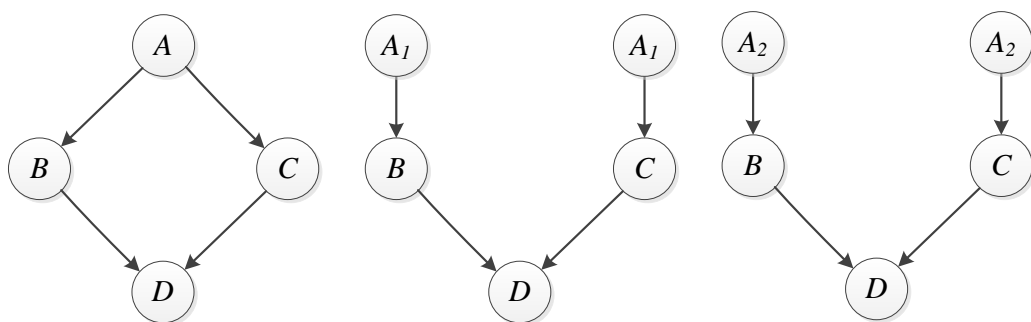


Рисунок 2.12 – Зображення зміни багатозв'язної мережі в однозв'язні [50]

Формулу (2.36) [50] обчислення сукупного імовірнісного розподілу для мережі можна записана таким чином:

$$\begin{aligned} P(A, B, C, D) &= \\ &= P(A = A_1, B, C, D) \cdot P(A = A_1) + P(A = A_2, B, C, D) \cdot P(A = A_2) \end{aligned} \quad (2.36)$$

Після певних перетворень отримані мережі перетворюються на об'єкти застосування алгоритму розповсюдження повідомлення для однозв'язних мереж Джуді Перла. Надалі результати для кожного із можливих варіантів інстанціювання мають бути комбіновані з ваговими коефіцієнтами у залежності від апіорних імовірностей.

Сама ступінь складності алгоритму знаходиться в залежності від кількості інстанціювань, що необхідно виконати й зростає експоненційно  $O(s \cdot n)$ , де  $s$  – це розмір визначеного перетину, а  $n$  – це є кількість станів, що зможуть приймати вершини перетину. Завдання із мінімізування розміру визначеного перетину є NP-складним.

#### 2.4.3 Алгоритм виключення змінних

Наприкінці минулого вже сторіччя з'явився алгоритм виключення змінних через процедуру їхнього почергового додавання [53].

Основна концепція цього алгоритму – обчислення імовірності вершини за наступною формулою (2.37) [53]:

$$P(X_n, e) = \sum_{x_k} \dots \sum_{x_3} \sum_{x_2} \prod_i P(X_i | Pa(X_i)), \quad (2.37)$$

що базується на формулі (2.38) [53] з декомпозиції сукупного розподілу імовірностей мережі:

$$\begin{aligned}
 P(X_1, \dots, X_n) &= P(X_n | X_1, \dots, X_{n-1}) \cdot P(X_{n-1} | X_1, \dots, X_{n-2}) \cdot \dots \cdot P(X_1) = \\
 &= \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) = \prod_{i=1}^n P(X_i | Pa(X_i)),
 \end{aligned}
 \tag{2.38}$$

де множина  $Pa(X_i)$  – це множина вершин-батьків вершини  $X_i$ .

Рис. 2.13 наочно зображає стрілочками процес знаходження імовірності вершини  $F$  на прикладі послідовного виключення вершин Мережі Байєса.

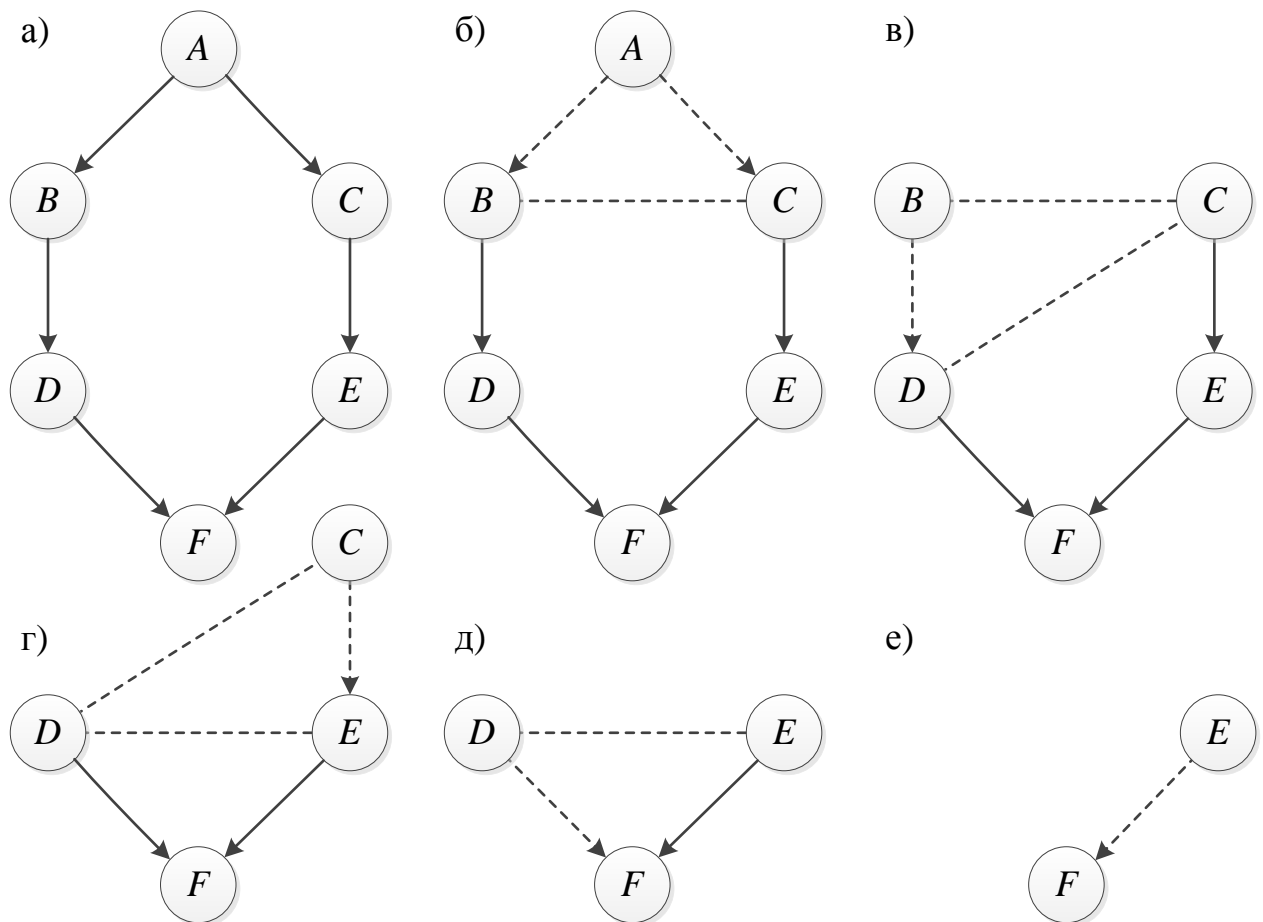


Рисунок 2.13 Схема прикладу виключення вершин [53]

Рис. 2.13 а) унаочнює початкову структуру мережі. Рис. 2.13 б) робить унаочненим виключення вершини  $A$ . Рис. 2.13 в) ілюструє виключення вершини  $B$  мережі, рис. 2.13 г) – вершини  $C$  мережі, рис. 2.13 д) – вершини  $D$  мережі та рис. 2.13 е) – вершини  $E$ . В кінцевому підсумку залишається лише тільки вершина  $F$ , яку й потрібно було відшукати.

Послідовність виконання обчислень для знаходження імовірності для даної мережі полягає в наступних кроках:

1. На першому кроці знаходимо сукупний розподіл щодо імовірності вершин  $A, B$  і  $C$  для даної мережі:  $P(A, B, C) = P(A) \cdot P(B | A) \cdot P(C | A)$ . Виключення вершини  $A$  мережі:  $P(B, C) = \sum_A P(A, B, C)$
2. Далі наступний крок – знаходимо сукупний розподіл імовірності вершин  $B, C$  і  $D$  для даної мережі:  $P(B, C, D) = P(B, C) \cdot P(D | B)$ . Виключення вершини  $B$  мережі:  $P(C, D) = \sum_B P(B, C, D)$
3. На подальшому кроці знаходимо сукупний розподіл імовірності вершин  $C, D$  і  $E$  для даної мережі:  $P(C, D, E) = P(C, D) \cdot P(E | C)$ . Виключення вершини  $C$  мережі:  $P(D, E) = \sum_C P(C, D, E)$
4. На цьому кроці схема дії алгоритму виключення змінних передбачає наступний етап – знаходження сукупного розподілу імовірності вершин  $D, E$  і  $F$  для даної мережі:  $P(D, E, F) = P(D, E) \cdot P(F | D, E)$ . Виключення вершини  $D$  мережі:  $P(E, F) = \sum_D P(D, E, F)$
5. Останнім етапом реалізації ідеї алгоритму передбачається виконання процедури виключення вершини  $E$  мережі:  $P(F) = \sum_E P(E, F)$

Величина, що визначає складність алгоритму, залежна від кількості операцій множення і додавання, які залежать в свою чергу вже від порядку виключення змінних. Завдання знаходження оптимального порядку виключення являється NP-складним.

Інший алгоритм 1996 року створення із назвою поглинаючого виключення (bucket elimination). В цьому, запропонованому Rina Dechter алгоритмі, ТУЙ конвертуються до відповідних  $\lambda$  – таблиць для розв'язку поставленої задачі [54].

## Висновки до розділу 2

В розділі 2 проведено аналіз принципів побудови МБ за навчальними даними та аналіз методів побудови ймовірнісного висновку в МБ. В результаті

проведеного аналізу було синтезовано необхідні висновки.

Мережі Байєса реалізують ідеї щодо використання імовірнісних графічних моделей подій, засновані на теорії імовірностей і теорії графів. Мережі Байєса заслужено вважаються переважним інструментом розв'язання двох актуальних прикладних наукових проблем, які дуже часто мають місце в прикладній математиці і техніці – проблема невизначеності та проблема складності [41].

Графічна наочність мережі Байєса є суттєвою перевагою для вибору саме цього наукового інструменту та унаочнює інтуїтивне розуміння досліджувачем проблеми або користувачем методу побудови структури вирішуваної проблеми і є розв'язком, коли постає питання складності, а імовірнісна складова цього надпотужного інструменту розв'язує актуальну проблему невизначеності в наявних даних мережі.

В результаті дослідження основних типів мереж Байєса, враховуючи дискретну природу вхідних даних, для використання у СППР обрано дискретні мережі байєса для побудови прогнозу курсу криптовалюти Біткоїн.

Як показало дослідження, основні проблеми, що виникають при побудові мереж Байєса за навчальними даними - це складність обчислень існуючих алгоритмів, необхідність побудови структури мережі, можливість ситуації неповних спостережень. Було описано евристичний метод побудови, обраний для вирішення проблем побудови структури мережі Байєса за навчальними даними, тому що для побудованої СППР характерна ситуація з повними спостереженнями та невідомою структурою мережі Байєса.

Було розглянуто методи побудови точного ймовірнісного висновку у мережах Байєса та обрано алгоритм виключення вершин, як той, що буде імплементований у розробленому методі прогнозування курсу Біткоїн через можливість використання його для багатозв'язних мереж Байєса.

## РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГНОЗУВАННЯ КУРСУ БІТКОІН

### 3.1 Опис програмного продукту

Для функціонування СППР, представленої в роботі, розроблено метод прогнозування криптовалюти Біткоїн. В якості вхідних використано статистичні дані понад 24 незалежних функцій, що стосуються торгівлі біткоїнами та мережі Біткоїн. Ці функції були обрані через їх тісний зв'язок з ціноутворенням Біткоїн. Опис зазначених даних наведено у таблиці 3.1. Дані складаються зі спостережень, що збираються щодня протягом останніх 10 років, з моменту появи Біткоїн. Інформація була отримана за допомогою ресурсу <https://www.quandl.com> [37]. 24-х годинний крок часового ряду мінімізує проблеми з шумом від вимірювань більш високої зернистості та хвилинної волатильності, а також служить для визначення конкретних аспектів, які були релевантними в прогнозуванні ціни криптовалюти Біткоїн. Ці функції включають такі поняття, як ринкова капіталізація Біткоїн, а також відношення обсягу трансакцій Біткоїн до обсягу USD. Метод розглядає різницю кожної з цих змінних, щоб передбачити ознаку зміни ціни, на відміну від самої фактичної ціни, щоб представити цю проблему біноміально, для застосування до прогнозу мережі Байєса. Для використання дискретної мережі Байєса також проводиться попередня обробка даних – дискретизація до 10 різних значень за принципом однакової кількості елементів. Навчальний набір даних складається з перших 70% цих даних, тоді як тестовий набір містить решту – 30%.

Таблиця 3.1 – Опис змінних для набору даних

Назва змінної	Опис
MKPRU	Ринкова ціна Біткоїн у USD
MWNUS	Біткоїн кількість користувачів My Wallet
TRVOU	Об'єм торгів з обміну Біткоїн на USD
DIFF	Біткоїн Difficulty
MWNTD	Біткоїн кількість користувачів My Wallet в день
MWTRV	Біткоїн об'єм транзакцій My Wallet в день
AVBLS	Біткоїн середній розмір блоку
BLCHS	Біткоїн розмір api.blockchain
MIREV	Біткоїн заробіток майнерів
HRATE	Біткоїн Hash Rate
CPTRA	Біткоїн ціна за транзакцію
CPTRV	Біткоїн відсоток ціни у всьому об'ємі транзакцій
ETRVU	Біткоїн оцінка об'єму транзакцій у USD
ETRAV	Біткоїн оцінка об'єму транзакцій
TOUTV	Біткоїн повний об'єм випуску
NTRBL	Біткоїн кількість транзакцій на один блок
NADDU	Біткоїн кількість використаних унікальних адрес
NTREP	Біткоїн кількість маловживаних адрес
NTRAT	Біткоїн повна кількість транзакцій
NTRAN	Біткоїн кількість транзакцій в день
TRFUS	Біткоїн повні оплата транзакцій у USD
TRFEE	Біткоїн повні оплата транзакцій
MKTCP	Біткоїн капіталізація ринку
TOTBC	Біткоїн всього на ринку
ATRCT	Біткоїн середній час підтвердження транзакції



Вхідний набір даних розбивається у відношенні 70:30 на навчальні дані та тестові дані. На 70% початкового набору даних проводиться побудова структури мережі Байєса евристичним методом та навчання параметрів мережі Байєса. Використовуючи алгоритм знаходження точного ймовірнісного висновку виключенням вершин будується прогноз для цільової вершини 30% початкового набору даних та формується нова колонка зі спрогнозованими значеннями вузла.

Таким чином у результаті двійкової класифікації утворюється біноміальна змінна що відображає прогноз зміни курсу крипто валюти Біткоїн на наступну добу – зростання або спад. Вона забезпечує роботу побудованої СППР з формування інвестиційної стратегії та зниження відповідних ризиків, представляючи особі, що приймає рішення, прогноз курсу криптовалюти Біткоїн на наступну добу.

В результаті імплементації методу була створена СППР для прогнозування зміни курсу криптовалюти Біткоїн, що складається з двох пов'язаних програмних модулів, як зображено на рис. 3.1. Перший модуль використовується для побудови мереж Байєса за навчальними даними, і написаний мовою програмування SAS Base в середовищі SAS 9.3 TS Level 1M2 [55]. Другий модуль використовує результати роботи модуля SAS Base для використання побудованої мережі з метою знаходження ймовірнісного висновку. Він використовує мову програмування Python 3.8.0 та може бути виконаний в середовищі JetBrains PyCharm Professional 2019.2.4 IDE. Модуль Python застосовує побудовану мережу Байєса для прогнозування курсу криптовалюти Біткоїн.

На рис. 3.1 проілюстрована структура розробленої СППР, яка представляє собою широкий комплекс засобів для аналізу даних.

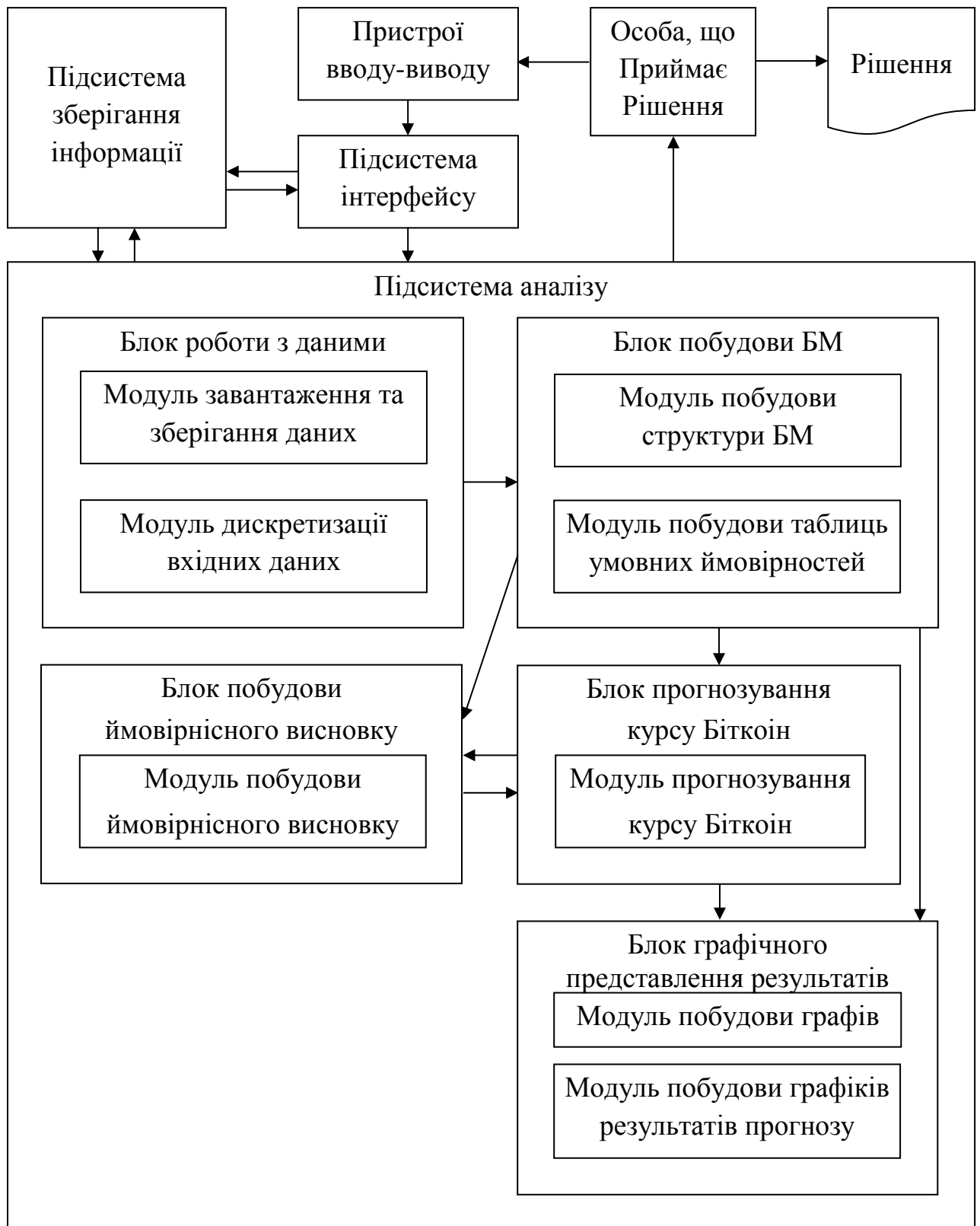


Рисунок 3.1 – СППР для прогнозування курсу криптовалюти Біткоїн

Користувач може завантажити дані в СППР, за допомогою пристроїв вводу-виводу. Підсистеми вводу-виводу зв'язані з підсистемою інтерфейсу програми.

Ця підсистема інтерфейсу функціонує як зв'язок користувача з внутрішніми підсистемами та елементами. Вона виводить інформацію для аналітиків та інвесторів, гравців на біржі. Також вона гарантує доступ до зовнішніх запам'ятовуючих пристроїв для вилучення та збереження необхідної інформації. Оператор вводить команди, дані, і запити в СППР та вихідні дані виводяться в зручному для сприйняття вигляді.

Підсистема збереження інформації представляє собою базу даних, що накопичує дані в ході роботи програми для їх подальшої обробки, аналізу, та використання. Побудова мережі Байєса реалізована за допомогою евристичного методу для знаходження структури мереж Байєса з допомогою навчальних даних. Підсистема виведення результатів являє собою набір таблиць та звітів для користувача, табличні репрезентації усіх проміжних кроків та висновків, таблицю зв'язності побудованої мережі, таблицю з результатом побудованого прогнозу. Для роботи програмного продукту необхідна наявність ПК, що задовольняє наступні вимоги:

- а) операційна система Windows;
- б) оперативна пам'ять розміром 2 Гбайт;
- в) вільний дисковий простір: 2 Гбайт для розміщення виконавчого файлу та середовища виконання, та додатковий простір (100 Мбайт) для вхідних даних і збережених результатів роботи;
- г) клавіатура;
- д) монітор з розподільчою здатністю 1024x768;
- ж) інсталяція SAS Studio Single-User або SAS OnDemand for Academics ([https://www.sas.com/en\\_us/software/on-demand-for-academics.html](https://www.sas.com/en_us/software/on-demand-for-academics.html) [56]);
- з) доступ до середовища розробки JetBrains PyCharm Professional 2019.2.4 IDE для Python 3.8.0 [57].

### 3.2 Інструкція з експлуатації програмного продукту

Інтерфейс є звичайним для програмного забезпечення, що працює на платформі SAS Studio. Структуру основної робочої панелі модуля проілюстровано на рис. 3.2.

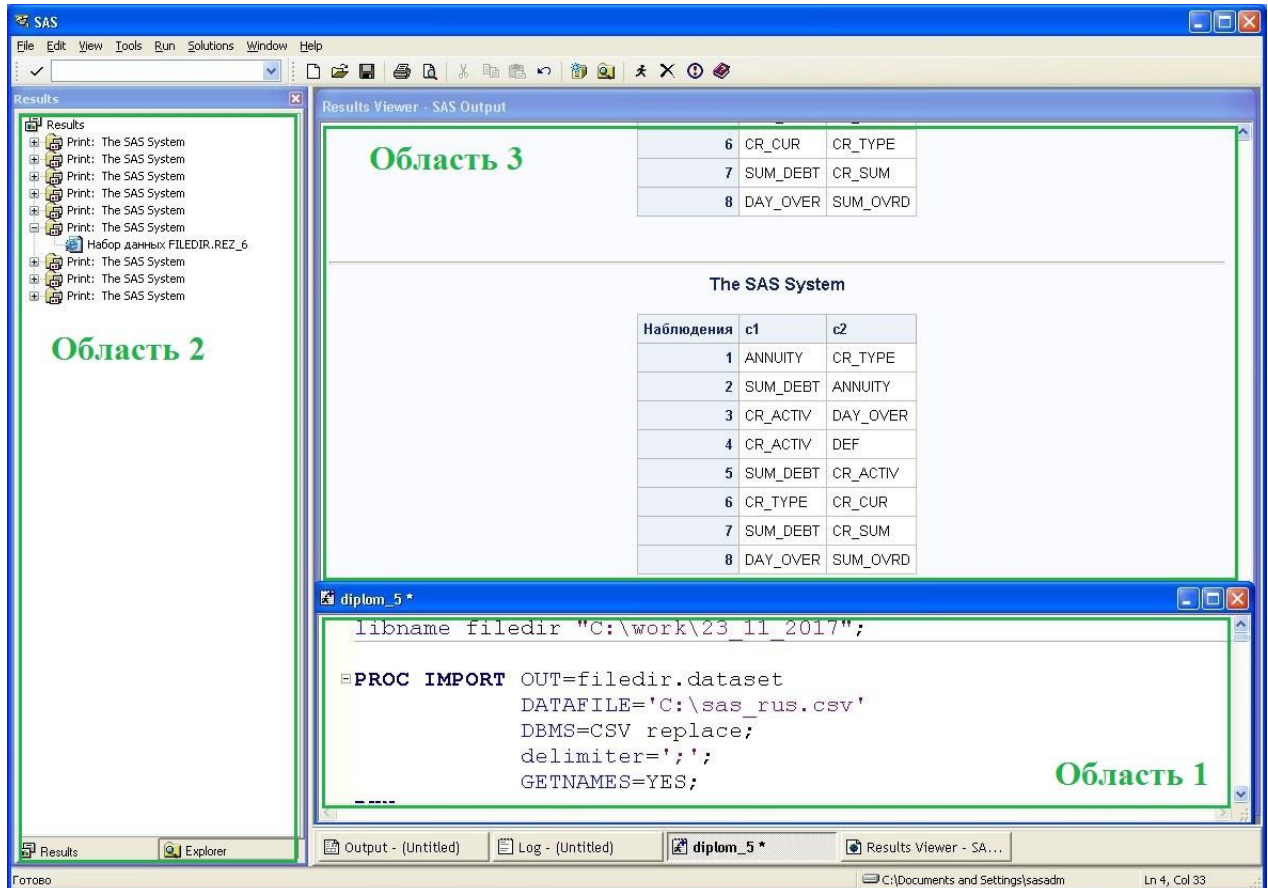


Рисунок 3.2 – Структура програмного модуля для побудови мереж

Головна форма програмного модуля містить три частини:

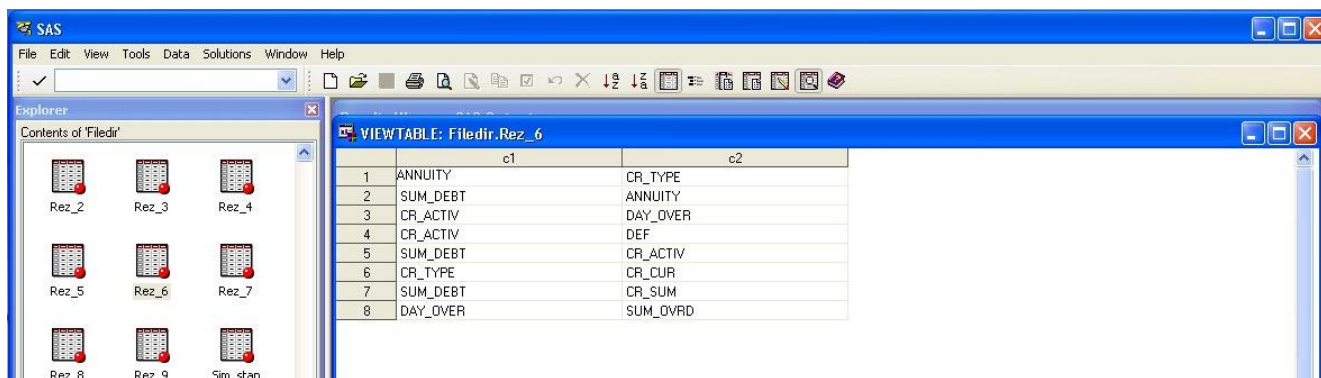
1. Редактор коду програми («Область 1» на рис. 3.2), тут користувач може вказати шлях до розташування вхідних даних на власному комп'ютері, тип файлу (.xlsx), шлях для збереження результатів роботи програми.
2. Дерево результатів («Область 2» на рис. 3.2), дозволяє аналіз, та обробку збережених під час роботи файлів та таблиць з проміжними і остаточними результатами.

3. Область відображення результатів («Область 3» на рис. 3.2), використовується виведення структури у вигляді матриці суміжності побудованого графу (опція «Results»).

Програмний модуль дозволяє обробляти набори даних необмеженої величини, що можуть бути подані в форматі .xlsx. Для попередньої підготовки даних для аналізу найкраще використовувати програми Microsoft Excel, SAS EM, SAS EG. Дані можуть подаватися як числові, так і строкові. Для того, щоб завантажити файл на обробку необхідно зазначити шлях його розташування в пам'яті комп'ютера. Основна функція програмного модуля полягає в побудові мережі Байєса, що використовує для роботи навчальні дані та евристичний метод. Для старту побудови моделі потрібно натиснути кнопку «Submit» в основному вікні програмного модулю.

Третя область відображає таблиці з результатами у форматі HTML, а також займається побудовою звіту. Таблиця є матрицею суміжності результуючого графу. Програма зберігає результати моделювання мережі та усі проміжні результати до бази даних для подальшого аналізу. Для їх перегляду необхідно відкрити на головній формі програмного модулю у області 2 необхідний файл, після цього вивести його на екран, як видно на рис. 3.3. Для цієї мети в області 2 потрібно скористатись опцією «Explorer».

Ресурс [https://csacademy.com/app/graph\\_editor/\[58\]](https://csacademy.com/app/graph_editor/[58]) використовується для візуалізації побудованої матриці суміжності у вигляді графу.



The screenshot shows the SAS Explorer window. On the left, there is a file explorer showing a directory structure with files named Rez\_2, Rez\_3, Rez\_4, Rez\_5, Rez\_6, Rez\_7, Rez\_8, Rez\_9, and Sim\_stan. On the right, a table titled 'VIEWTABLE: Filedir.Rez\_6' is displayed, showing a matrix of results with columns c1 and c2.

	c1	c2
1	ANNUITY	CR_TYPE
2	SUM_DEBT	ANNUITY
3	CR_ACTIV	DAY_OVER
4	CR_ACTIV	DEF
5	SUM_DEBT	CR_ACTIV
6	CR_TYPE	CR_CUR
7	SUM_DEBT	CR_SUM
8	DAY_OVER	SUM_OVRD

Рисунок 3.3 – Матриця суміжності побудованої мережі

Даний програмний модуль включає в себе ряд підмодулів та процедур, що забезпечують його роботу: обробка вхідних даних, підрахунок імовірностей

вершин, оцінка значень  $MI$ , оцінка значень  $H$ , ранжування вершин за значенням  $MI$ , оцінка значень  $\alpha$ , побудова структур мереж, оцінка ациклічності графа, формування результатів. Структура даного модуля забезпечує послідовне їх виконання в циклі, доки усі передбачені алгоритмом структури не будуть побудовані, та порівняні між собою за критерієм оптимальності, та одна з них обрана як результуюча структура, а її матриця суміжності подана як результат.

Модуль побудови прогнозу курсу криптовалюти Біткоїн

Інтерфейс є звичайним для програмного забезпечення, що працює на платформі JetBrains PyCharm IDE. Структуру основної робочої панелі модуля проілюстровано на рис. 3.4.

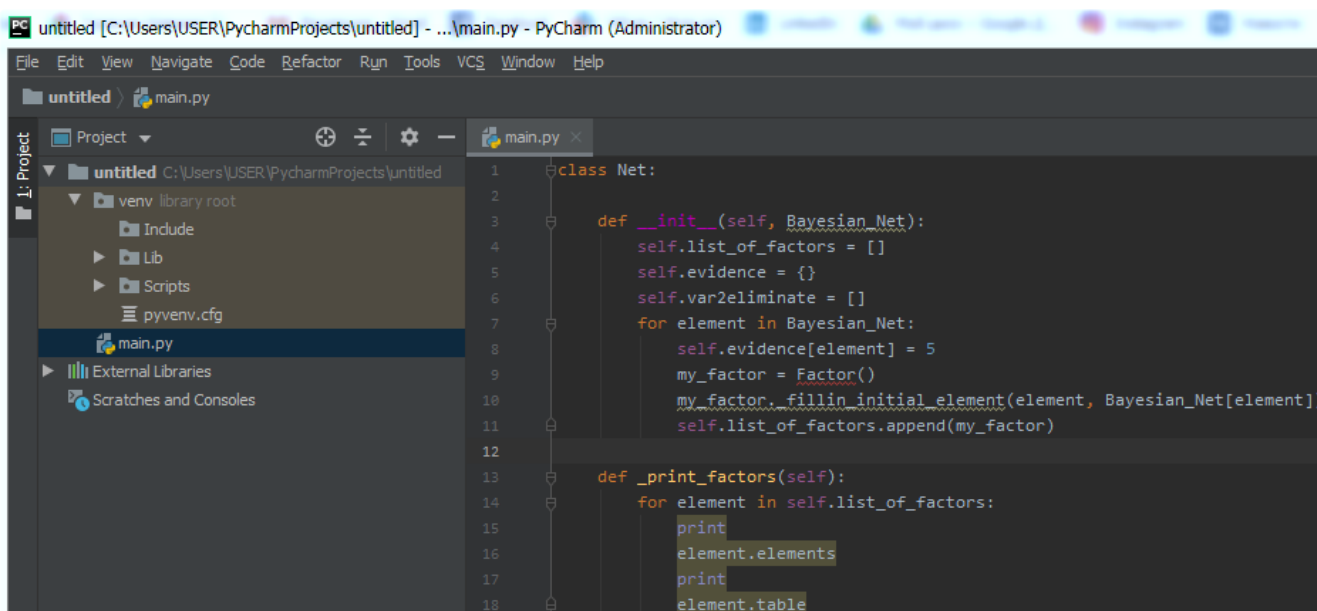


Рисунок 3.4 – Програмний модуль побудови прогнозу курсу криптовалюти Біткоїн

IDE представляє собою редактор коду програми. Перед запуском програмного модуля необхідно задати шлях до папки з результатами роботи модуля SAS Base – набором даних, матрицею суміжності, представленими у вигляді файлів з розширенням .xlsx. Для початку роботи модуля необхідно вибрати пункт Run->Run, як зображено на рис. 3.5, або комбінацією клавіш Alt+Shift+F10.



fillin\_initial\_element(self, key, value), set\_evidence(self, element2set, value), build\_factor(self, factors\_involved), var2address(self, variables2check), ddress2var(self, address), define\_variables(self, by\_hand = True, code = None), def \_set\_evidence\_net(self), calculate\_cost(self, element), choose\_var2eliminate(self), eliminate\_variable(self, element), mult\_factors(self) variable\_elimination(self), що забезпечують функціонування алгоритму.

Після навчання параметрів мережі за навчальними даними програма переходить до процедури знаходження точного ймовірнісного висновку для обраної підмножини вершин. Якщо запитується лише одна змінна, програма виведе ймовірність цієї змінної. Якщо запитується більше одного, програма виведе таблицю, що описує спільний розподіл ймовірності всіх запитуваних змінних.

В алгоритмі усунення змінних ми проводимо усунення змінних, комбінуючи фактори, що містять цю змінну. Наприклад, якщо ми хочемо усунути  $b$  шляхом комбінування факторів  $F1(a, b)$  і  $F2(b, c)$ , нам потрібно створити фактор  $F3(a, b, c)$  і маргіналізувати над  $b$ , щоб отримати  $F3(a, c)$ . Складність алгоритму усунення змінних становила  $n \cdot m^k$ , будучи  $n$  числом змінних,  $m$  – кількістю значень кожної змінної і  $k$  максимальним розміром фактора, створеного до маргіналізації.

На жаль, вибір правильного впорядкування змінних для усунення також є NP-складною проблемою. На практиці існують евристички, які дають досить хороші показники. У цій реалізації імплементована політика Min-neighbour, яка складається з вибору змінної з менш залежними змінними. Оскільки після об'єднання факторів нові чинники вже не є розподілом ймовірностей, розглядається непрямуювана графічна модель і обчислюються залежні змінні як ті, що поділяють будь-який фактор.



### 3.3 Результати тестування програмного продукту

Для використання початкового набору даних для прогнозування була виконана попередня обробка даних. Значення кожної змінної були замінені на різницю між сьогоднішнім значенням та учорашнім – для прогнозування не самого курсу Біткоїн, а скоріше його динаміки. Для застосування в якості інструменту прогнозування мережі Байєса була виконана дискретизація отриманого набору даних. Цільова вершина результуючого набору даних приймає значення -1 або 1 в залежності від росту чи падіння курсу у відповідний день. Інші змінні приймають кожна 10 різних значень. Дискретизацію було проведено засобами SAS EM за принципом однакової кількості елементів (додаток Б).

Було проведено три експерименти для дослідження ефективності роботи прогнозування на різних історичних періодах розвитку Біткоїн: 2009-2014 рр, 2015-2019 рр, 2009-2019рр.

В результаті роботи програмного модуля для побудови мереж Байєса було отримано структуру, зображену на рис. 3.6. Для візуалізації результатів було використано ресурс [https://csacademy.com/app/graph\\_editor/](https://csacademy.com/app/graph_editor/)



Таблиця 3.2 – результати прогнозу для 2015-2019 рр.

		Спрогнозовані	
		-1	1
Дійсні	-1	223	3
	1	49	267

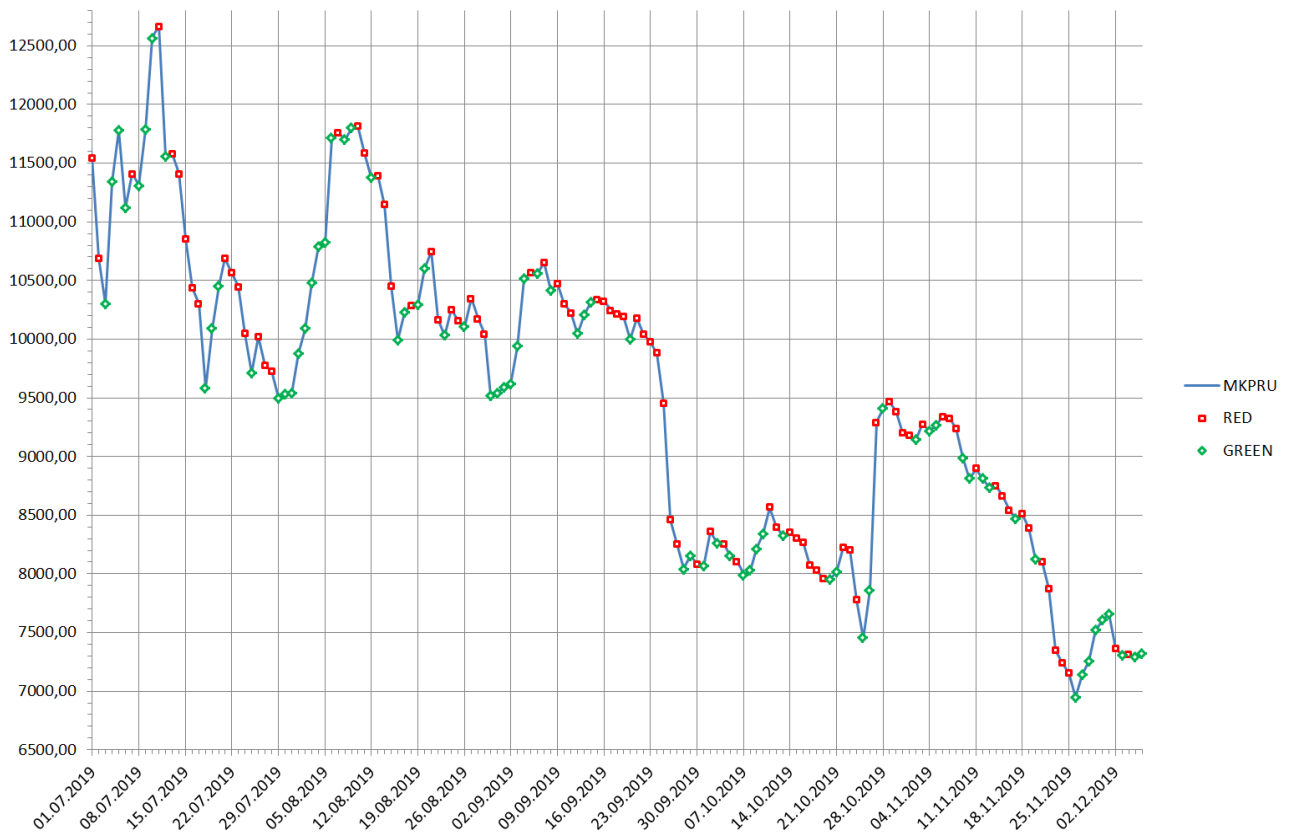


Рисунок 3.7 – Результати прогнозу для 2015-2019 рр.

В результаті експерименту з часовим проміжком 2009-2014 рр. було отримано наступні результати (табл. 3.3): Sensitivity (TPR) становила 0,84558824, Specificity (TNR) становила 0,99256506, Precision (PPV) становила 0,99137931 Accuracy (ACC) становила 0,91866913. Задовільну точність результатів можна віднести до тривалого часового інтервалу між точками даних, що призводять до зменшення коливань цін на фактичному ринку біткоіна а також великого розміру вибірки для навчання мережі Байеса. Результати роботи методу прогнозування криптовалюти Біткоін зображено на рис. 3.8.

Таблиця 3.3 – результати прогнозу для 2009-2014 рр.

		Спрогнозовані	
		-1	1
Дійсні	-1	230	2
	1	42	267

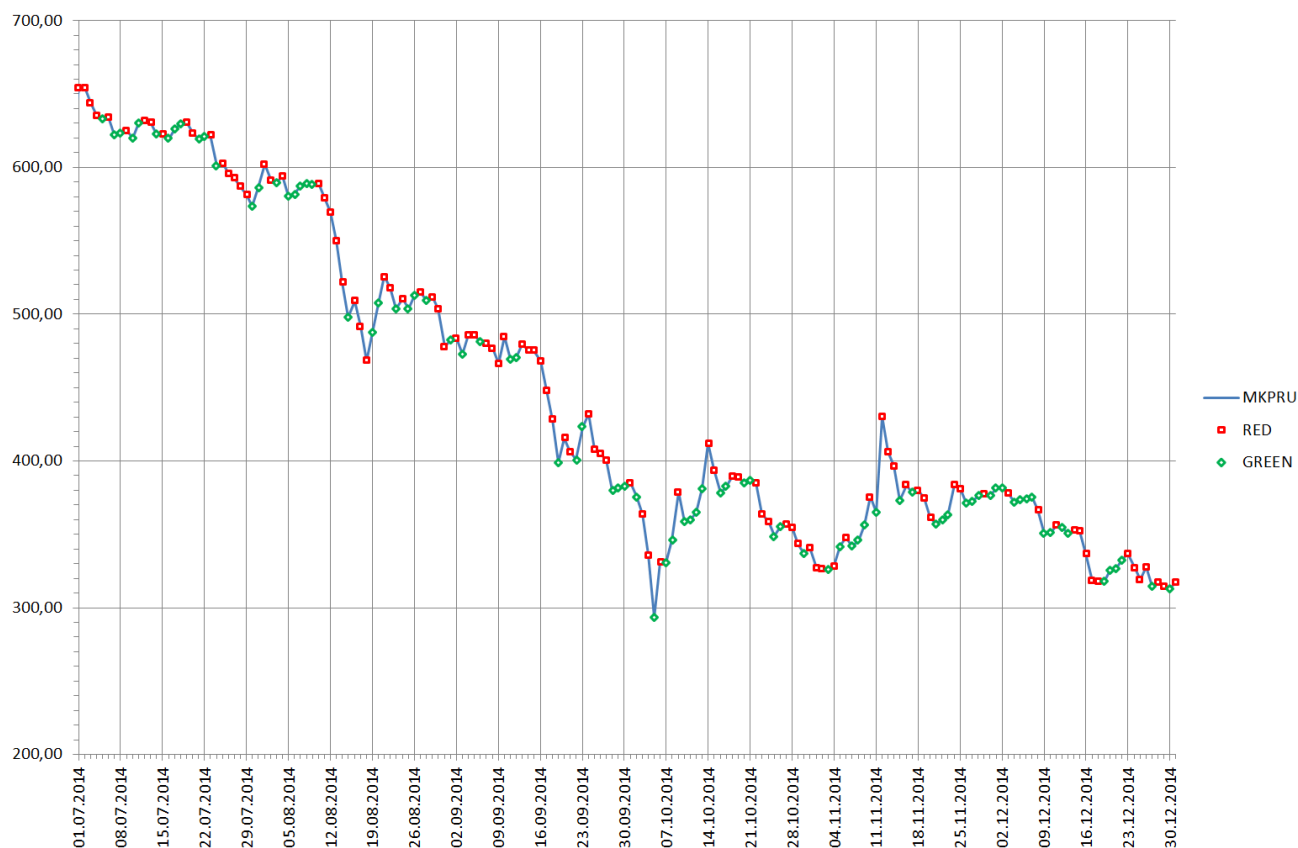


Рисунок 3.8 – Результати прогнозу для 2009-2014 рр.

В результаті експерименту з часовим проміжком 2009-2019 рр. було отримано наступні результати (табл. 3.4): Sensitivity (TPR) становила 0,81985294, Specificity (TNR) становила 0,98888889, Precision (PPV) становила 0,98672566, Accuracy (ACC) становила 0,90405904. Задовільну точність результатів можна віднести до тривалого часового інтервалу між точками даних, що призводять до зменшення коливань цін на фактичному ринку біткоіна а також великого розміру вибірки для навчання мережі Байєса. Результати роботи методу прогнозування криптовалюти Біткоін зображено на рис. 3.9.

Таблиця 3.4 – результати прогнозу для 2009-2019 рр.

		Спрогнозовані	
		-1	1
Дійсні	-1	477	5
	1	97	565



Рисунок 3.9 – Результати прогнозу для 2009-2019 рр.

Для порівняння якості побудованого прогнозу в програмному продукті з використанням мереж Байєса, було проаналізовано роботу Isaac Madan, Shaurya Saluja та AoJia Zhao [36]. У розглянутій роботі було проведено прогноз зміни курсу криптовалюти Біткоїн, засновуючись на щоденних вимірах. Кількість досліджуваних в роботі функцій – 26. Проміжок часу – 2009-2014 рр. Використані методи прогнозування – біноміальна логістична регресія, метод опорних векторів та алгоритм випадкових лісів [36]. Результати порівняння відображені в табл. 3.5.

Таблиця 3.5 – Порівняльний аналіз якості прогнозування

	Отримані результати			Isaac Madan, Shaurya Saluja та Aojia Zhao		
Статистика	2009-2014	2015-2019	2009-2019	BINOMIAL GLM	SVM	RANDOM FOREST
Sensitivity (TPR)	0,84558824	0,83101045	0,81985294	0.9790	0.0348	1.0000
Specificity (TNR)	0,99256506	0,99122807	0,98888889	0.9939	0.5514	0.9392
Precision (PPV)	0,99137931	0,98962656	0,98672566	0.9790	0.0839	0.7762
Accuracy (ACC)	0,91866913	0,91083916	0,90405904	0.9879	0.2716	0.9498

Отримано результати виконання трьох різних прогнозів на щоденних даних Біткоїн різних періодів, використовуючи 24 характеристики, описані вище, як показано на рис. 3.10.

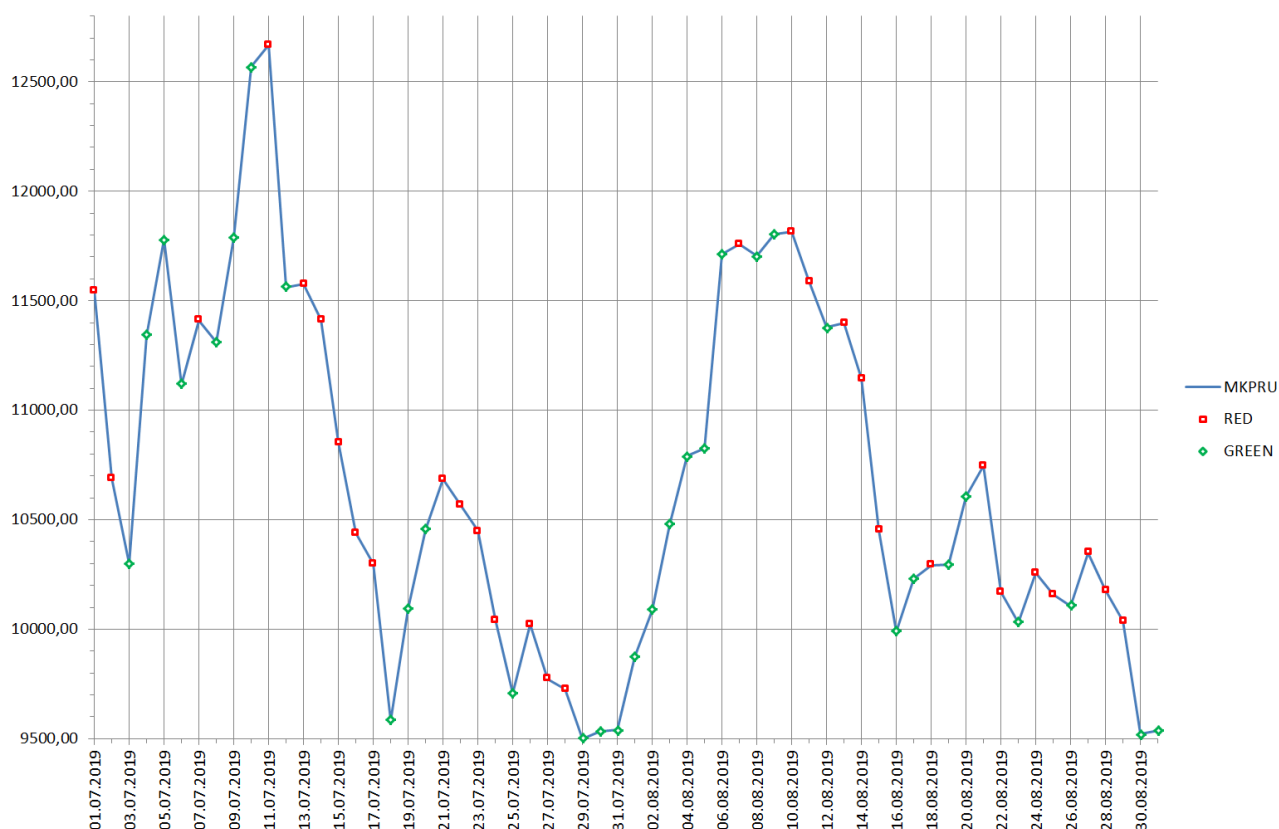


Рисунок 3.10 – Графік частини побудованого прогнозу

Задовільну точність результатів можна віднести до тривалого часового інтервалу між точками даних, що призводять до зменшення коливань цін на фактичному ринку біткоїна. Аналіз табл. 3.5 показує, що отримана точність є задовільною для прогнозування курсу криптовалюти Біткоїн. Результати

власного прогнозу (Accuracy 91.9%) наближаються за точністю до результатів, отриманих в роботі Isaac Madan, Shaurya Saluja та Aojia Zhao [36] у методах біноміальної логістичної регресії (Accuracy 98.8%) та алгоритму випадкових лісів (Accuracy 95.0%), та перевищує за точністю результат методу опорних векторів (Accuracy 27.2%). Меншу точність власного методу можна частково пояснити меншою кількістю досліджуваних змінних у навчальних даних. Таким чином, проведені експерименти можна вважати успішними, на розглянутих часових інтервалах СППР виконала завдання з прогнозування курсу криптовалюти Біткоїн.

### Висновки до розділу 3

В розділі 3 було виконано реалізацію евристичного алгоритму побудови МБ у вигляді програмного продукту та ймовірного висновку в побудованій мережі. Програмний продукт апробовано на навчальних даних, що становлять вибірку історичних даних коливання курсу криптовалюти Біткоїн. Проведено аналіз ефективності створеної моделі, порівняння точності побудованого прогнозу з результатами інших досліджень [36] у сфері прогнозування курсу криптовалют.

Було досягнуто мети даної роботи: була створена СППР для прогнозування зміни курсу криптовалюти Біткоїн, що складається з двох пов'язаних програмних модулів. Для тестування програмного продукту на реальних даних була використана вибірка щоденних даних стосовно ціни Біткоїн в USD та додатковими 24 функціями мережі та ринку Біткоїн. Було проведено три експерименти для дослідження ефективності роботи прогнозування на різних історичних періодах розвитку Біткоїн: 2009-2014 рр, 2015-2019 рр, 2009-2019рр. Результати порівняні з іншими дослідженнями в сфері прогнозування курсу криптовалюти Біткоїн. Прогнозування проведено з точністю більше 90%. Задовільну точність результатів можна віднести до тривалого часового інтервалу між точками даних, що призводять до зменшення

коливань цін на фактичному ринку біткоіна. Порівняльний аналіз отриманих результатів показав, що отримана точність є задовільною для прогнозування курсу криптовалюти Біткоїн.



## РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

В останні роки такий вид малого підприємництва як стартап набув великої популярності. Такий комерційний проект звичайно знаходиться в стані розробки, або нещодавно вийшов на ринок. Характерною особливістю стартапу, на відміну від малого бізнесу, є оригінальність та інновації, він не може бути копією вже реалізованих ідей. При цьому проект не обов'язково повинен бути масштабним, головне, щоб він був креативним, а його завдання – спрощувати людям будь-які дії в їх повсякденному житті.

Наразі, з появою Інтернету та сучасних технологій, стало простіше знаходити відповідну нішу на ринку, знаходити інвесторів та споживачів. З'явилося набагато більше можливостей для розвитку свого проекту за кордоном, ніж раніше. Проте розробка стартапу пов'язана з досить високими ризиками. Не всім вдається довести свій стартап-проект до впровадження на ринку. За статистикою успіху досягає лише 10-20% від усіх стартап-проектів.

Запуск стартапу передбачає цілий ряд обов'язкових дій, в межах яких визначають ринкові перспективи стартапу, графік розробки, принципи організації виробництва, заходи з залучення інвесторів та аналіз ризиків.

### 4.1 Опис ідеї проекту

У таблиці 4.1 подано зміст ідеї стартап-проекту, можливі напрямки застосування та основні вигоди, що може отримати користувач товару. У таблиці 4.2 визначені сильні, слабкі та нейтральні сторони проекту.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмний продукт для прогнозування курсу криптовалюти Біткоїн на основі застосування мереж Байєса	Біржі обміну криптовалют	Дозволяє користувачам з різним рівнем підготовки проводити необхідну попередню обробку даних для побудови прогнозуючої моделі, будувати мережу Байєса та одержувати прогнозні дані на основі побудованої моделі

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			
		Мій проект	BayesiaLab	GeNIe	SAS Enterprise Miner
1.	Ціна	Низька	Середня	Середня	Висока
2.	Функціонал	Вузький	Вузький	Широкий	Широкий

Отже, з табл. 4.2 можна визначити, що ціна є сильною характеристикою для потенційного товару, а функціонал, зважаючи на напрямки застосування товару, є нейтральною властивістю.

#### 4.2 Технологічний аудит ідеї проекту

За результатами аналізу таблиці 4.3 можна зробити висновок про можливість технологічної реалізації проекту.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Програмний продукт для прогнозування курсу криптовалюти Біткоїн	Прогнозування на основі побудови статичних мереж Байєса	Наявна	Доступна
2	на основі застосування мереж Байєса	Прогнозування на основі побудови динамічних мереж Байєса	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: прогнозування на основі побудови динамічних мереж Байєса				

#### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартапу

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	1 000 ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	15%

За результатами аналізу таблиці 4.4 можна зробити висновок, що ринок є привабливим для входження за попереднім оцінюванням.

Визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці груп клієнтів	Вимоги споживачів
1	Прийняття рішення щодо інвестиції у портфель криптовалют	Біржі обміну криптовалют	Відмінність сфер діяльності клієнтів (короткочасні, довгострокові інвестування)	Висока точність прогнозування. Простий у використанні. Швидкодія при обробці значного об'єму інформації

Проведемо аналіз ринкового середовища: таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6-4.7).

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Наявність великої конкуренції	Вихід на ринок великої компанії	Вихід з ринку. Обрати нову цільову аудиторію. Передбачити переваги продукту, щоб повідомити про них саме після виходу великої компанії на ринок
2	Зміна потреб користувачів	Користувачам необхідні рішення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до продукту

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Відсутність конкуренції	Відсутність аналогічних продуктів для користувача на вітчизняному ринку	Локалізація та адаптація сервісу для локальних груп. Адаптація до вітчизняних особливостей
2	Поява нових цільових груп клієнтів	Потреба в аналогічному продукті в інших сферах діяльності	Адаптація продукту під нові сфери використання

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку (табл. 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополістична	Існує декілька фірм-конкурентів	Підтримка якості продукту та постійні вдосконалення
2. За рівнем конкурентної боротьби - інтернаціональний	Фірми конкуренти з різних країн	Підтримувати продукт на національному ринку
3. За галузевою ознакою - внутрішньогалузева	Продукт використовується в одній галузі	Вдосконалювати продукт для застосування в інших галузях
4. Конкуренція за видами товарів: - товарно-родова	Присутня конкуренція з боку товарів-замінників	Розширювати функціонал продукту
5. За характером конкурентних переваг - нецінова	Вдосконалення якості продукції, технології виробництва, інновацій	Випускати нові товари, які принципово відрізняються від своїх попередників та представляють модернізований варіант старої моделі
6. За інтенсивністю - немарочна	Роль торгової марки незначна	Приділяти увагу якості продукту а не бренду компанії

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	GeNie	SAS Enterprise Miner	Диференціація витрат, розширення каналів збуту	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу
Висновки :	Середня конкурентна боротьба з вже існуючими на ринку гравцями	Є можливості виходу на ринок, але є і конкуренти. Строки – пів року.	Постачальники не диктують умови роботи	Клієнти диктують умови роботи на ринку	Обмеження для роботи на ринку через товари-замінники

На основі аналізу конкуренції (табл. 4.9), а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6-4.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності (табл. 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Більш доступна ціна збільшує кількість потенційних клієнтів
2	Функціонал	Функціонал направлений на предметну область

За визначеними факторами конкурентоспроможності (табл. 4.10) проведемо аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін «SAB Analytical Studio»

№ п/п	Фактор конкурентоспроможнос ті	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з “Bayesian Network Creator”						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	18		+					
2	Функціонал	10					+		
3	Зручний інтерфейс	12				+			

Складемо SWOT-аналіз (матриця аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities)) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: ціна, зручний інтерфейс	Слабкі сторони: функціонал
Можливості: Низька конкуренція, поява нових потреб споживачів	Загрози: Висока конкуренція, не відповідність потребам споживачів

На основі SWOT-аналізу визначимо альтернативи виведення проекту на ринок та оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (табл. 4.13).



Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Імовірність отримання ресурсів	Строки реалізації
1	Створення програмного забезпечення	80%	4 місяці
2	Створення веб-сервісу	60%	7 місяців

#### 4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Біржі обміну криптовалют	Висока	Високий	Середня	Середня складність
2	Приватні інвестори	Середня	Середній	Помірна	Висока складність
Які цільові групи обрано: 1					

Для роботи в обраних сегментах ринку сформуємо базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспромо жні позиції	Базова стратегія розвитку*
1	Надання товару важливих з точки зору споживача властивостей, які роблять товар відмінним від товарів конкурентів	Визначити потреби кожної з цільових груп, розробити стратегії приваблення споживачів та маркетингові комунікації	Оперативне реагування на зміни в ринковому попиті, орієнтованість на кінцевого споживача, висока якість продукту	Стратегія диференці ації

Оберемо стратегію конкурентної поведінки (табл. 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідц ем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентн ої поведінки*
1	Не є першопрохідце м	Шукати нових	Ні	Стратегія заняття конкурентн ої ніші

Сформуємо ринкову позицію, за якою споживачі мають ідентифікувати проект(табл. 4.17).

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто- спроможні позиції власного проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Простий та зручний користувацький інтерфейс, надійність та безпека, швидкість роботи продукту	Стратегія диференці ації	Позиція на основі порівняння продукту компанії з продуктами конкурентів. Відмінні особливості споживачів	Автоматизація робочих процесів, зниження інвестиційних ризиків, зниження навантаження та часу

#### 4.5 Розроблення маркетингової програми стартап-проекту

У табл. 4.18 підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Автоматизація робочих процесів	Продукт автоматизує такі процеси, як обробка даних та прийняття рішення щодо інвестиції у портфель криптовалют	Після впровадження продукту прийняття рішення стає можливим на основі відсоткового передбачення імовірностей зміни курсу
2	Зменшен ня ризиків інвестиці й	Продукт зменшує інвестиційні ризики учасників біржі обміну криптовалют	Висока точність прогнозування знижує інвестиційні ризики учасників біржі обміну криптовалют
3	Зниження навантаж ення та часу	Продукт знижує навантаження на персонал учасників біржі обміну криптовалют та зменшує час прийняття рішення	Персоналу учасників біржі обміну криптовалют не потрібно самотійно аналізувати великий об'єм даних, що знижує навантаження на прискорює роботу

Розроблена трирівнева маркетингова модель товару(табл. 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмний продукт для прогнозування кредитоспроможності фізичних осіб. Повинен бути зручним, швидким та безпечним		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Попередня обробка даних	Нм	Тх/Тл
	2. Побудова мережі Байєса	Нм	Тх/Тл/Е
	3. Прогнозування курсу криптовалюти	Нм	Тх/Тл
	Якість: проходження тестування		
	Пакування: відсутнє		
	Марка: “ Bayesian Network Creator ”		
III. Товар із підкріпленням	До продажу: демонстрація демо-можливостей		
	Після продажу: навчання персоналу, супровід, технічна підтримка		
Вихідний код програмного продукту є закритим, та не передається клієнтам і третім особам. На програмний продукт оформлено авторське право			

Визначимо цінові межі, якими необхідно керуватись при встановленні ціни на товар (табл. 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1500\$	2000\$	Високий рівень доходів	Базова покупка та впровадження: нижня межа - 500\$, верхня межа - 1000\$.

Визначимо оптимальну систему збуту (табл. 4.21).

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – банківські учасників біржі обміну криптовалюти, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть автоматизувати робочі процеси. Вони цікавляться інноваційними рішеннями, відвідують тематичні онлайн конференції	Формування попиту і стимулювання збуту. Встановлення контактів із споживачами. Просування маркетингової інформації	Нульова або однорівнева (сервіс безпосередньо продається споживачам та через посередників)	Прямий канал збуту до споживача, мінімізувати витрати на додаткові канали збуту

Розроблена концепція маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуютьс я цільові клієнти	Ключові позиції, обрані для позиціонув ання	Завдання рекламного повідомлен ня	Концеп ція реклам ного зверне ння
1	Цільові клієнти – учасники біржі обміну криптовалют, що займаються інвестуванням, і бажають автоматизувати процес прийняття рішень, та зменшити кількість збиткових вкладень. Вони цікавляться інноваційними рішеннями, відвідують онлайн семінари	Конференції, форуми, новини у сфері інноваційних технологій, періодичні видання у професійних галузях, персональні й маркетинг	Позиція на основі порівнянн я продукту компанії з продуктам и конкуренті в. Відмінні особливос ті споживачі в	- інформуват и про новий продукт та його переваги; - сформувати сприятливу думку; - сформувати образ марки та її виробника у свідомості споживачів; - збільшити потік покупців	Зменш уємо інвести ційні ризики. Приско рюємо та автома тизуєм о процес прийма ття рішенн я про вкладе ння

## Висновки до розділу 4

В даному розділі проведено аналіз створення та виведення на ринок стартап-проекту на основі програмного продукту, який було розроблено в рамках магістерської дисертації.

В межах цього аналізу було розроблено опис самої ідеї проекту, визначено загальні напрями використання товару, проаналізовано ринкові можливості щодо впровадження проекту, визначено відмінності від конкурентів та розроблено стратегію виходу на ринок.

Узагальнюючи проведений аналіз, можна зазначити, що є можливість ринкової комерціалізації проекту. Наявний попит, динаміка ринку зростає. З огляду на потенційні групи клієнтів, а саме банківські установи, та високий рівень конкурентоспроможності проекту, є достатні перспективи для впровадження стартапу. Отже, подальша імплементація проекту є доцільною.



## ВИСНОВКИ

У роботі були вирішені усі поставлені задачі та досягнуто мети дослідження: побудовано СППР для прогнозування курсу криптовалюти Біткоїн на основі МБ за попередньо накопиченим масивом фактичних даних та показників, з метою мінімізування інвестиційних ризиків.

В розділі 1 було проведено аналіз принципів функціонування та ціноутворення криптовалюти Біткоїн, методи, що використовуються для прогнозування курсу Біткоїн.

Було проаналізовано концепцію технології блокчейн, яку дана технологія має реалізацію, які властивості її є важливими для прогнозування курсу Біткоїн. Проведено докладне дослідження технології блокчейн як основи для криптовалюти Біткоїн, які криптографічні алгоритми вона використовує, які будівельні блоки присутні у її структурі, провели аналіз ланцюгів блоків.

В розділі 2 було досліджено основні типи мереж Байєса та обрано дискретні мережі байєса для побудови прогнозу курсу криптовалюти Біткоїн. Вони були обрані через дискретну природу вхідних даних.

Було розглянуто основні проблеми, що виникають при побудові мереж Байєса за навчальними даними та описано евристичний метод побудови, що був обраний для вирішення проблем, що розв'язуються цією роботою.

Було розглянуто методи побудови точного ймовірнісного висновку у мережах Байєса та обрано алгоритм виключення вершин, як той, що буде імплементований у розробленому мутоді прогнозування курсу Біткоїн.

В розділі 3 виконано реалізацію евристичного алгоритму побудови МБ у вигляді програмного продукту та реалізацію ймовірнісного висновку в побудованій мережі. Програмний продукт апробовано на навчальних даних, що становлять вибірку історичних даних коливання курсу криптовалюти Біткоїн. Було досягнуто мети даної роботи: була створена СППР для прогнозування зміни курсу криптовалюти Біткоїн, що складається з двох пов'язаних програмних модулів. Для того, щоб забезпечити здатність обробляти великі

обсяги даних були використані сучасні мови програмування SAS Base та Python. Роботу програми перевірено в трьох експериментах з реальними даними. Досягнуто точності прогнозу 92%. Проведено порівняльний аналіз результатів прогнозування з іншими дослідженнями методів прогнозування змін курсу криптовалюти Біткоїн.

В четвертому розділі приведено аналіз створення й виведення на ринок реалізованого стартап-проекту на основі розробленого програмного продукту.

В межах наведеного аналізу був розроблений опис інноваційної ідеї актуального конкурентоспроможного проекту, визначені загальні напрями використання високотехнологічного товару, проаналізовані ринкові можливості впровадження проекту, визначені переваги над аналогічними продуктами конкурентів та розроблена стратегія виходу на ринок.

Узагальнюючи проведений аналіз, можна впевнено стверджувати, що є можливість ринкової комерціалізації реалізованого проекту.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Технология распределенного реестра Блокчейн (Blockchain) URL: <https://mining-cryptocurrency.ru/blockchain/> (дата звернення: 11.10.2019).
2. Технологія блокчейн та страхування URL: <http://www.insa.com.ua/uk/blog/tehnologiya-blockchain-i-strahovanie/> (дата звернення: 16.10.2019).
3. Что нам стоит блокчейн построить? URL: <https://habr.com/ru/post/443282/> (дата звернення: 15.10.2019).
4. Пол Винья, Майкл Кейси. Эпоха криптовалют. Как биткоин и блокчейн меняют мировой экономический порядок. 2017. С. 316–339.
5. Что такое блокчейн URL: <https://prostocoin.com/blog/blockchain-guide> (дата звернення: 15.10.2019).
6. Пол Винья, Майкл Кейси Машина правды. Блокчейн и будущее человечества. Манн, Иванов и Фербер, 2018. С. 127–143.
7. Andreas M. Antonopoulos Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly Media. December 23, 2018. P. 325–401.
8. Біткоїн Wiki URL: <http://ru.Біткоїнwiki.org> (дата звернення: 17.10.2019).
9. Биткоин и технологии криптовалюты URL: <https://www.intuit.ru> (дата звернення: 17.10.2019).
10. Joseph J. Bambara Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions. McGraw-Hill Education. February 16, 2018 P. 316–339.
11. Imran Bashir Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained. Packt Publishing. March 30, 2018 P. 574–586.
12. Біткоїн-терміни URL: <https://Біткоїн.org/uk/vocabulary> (дата звернення: 17.10.2019).
13. Bitcoin: Інновації у платіжних системах URL: <https://bitcoin.org/uk/innovation> (дата звернення: 10.10.2019).

14. Майнінг. Процес майнінгу URL: <https://pingblockchain.com/majning-proces-majningu/> (дата звернення: 18.10.2019).

15. Мыльный пузырь или выгодная инвестиция: почему дорожает биткоин URL: <https://www.dw.com/ru/мыльный-пузырь-или-выгодная-инвестиция-почему-дорожает-биткоин/a-41587949> (дата звернення: 18.10.2019).

16. Звідки взявся Біткоїн і в чому його переваги URL: <https://nv.ua/ukr/techno/technoblogs/zvidki-vzjavsja-bitkoini-i-v-chomu-jogo-perevagi-blog-sergija-kalinina-2129401.html> (дата звернення: 14.10.2019).

17. Bitcoin: основные принципы и отдельные юридически-значимые особенности URL: [https://protocol.ua/ua/bitcoin\\_osnovnie\\_printsipi\\_i\\_otdelnie\\_yuridicheski\\_znachimie\\_osobennosti/](https://protocol.ua/ua/bitcoin_osnovnie_printsipi_i_otdelnie_yuridicheski_znachimie_osobennosti/) (дата звернення: 07.10.2019).

18. Выгодный майнинг. Почему добыча Bitcoin никогда не станет убыточной URL: <https://www.rbc.ru/crypto/news/5cad91fc9a79472c5f1670fd> (дата звернення: 14.10.2019).

19. Andreas M. Antonopoulos Mastering Bitcoin: Programming the Open Blockchain. O'Reilly Media. July 1, 2017. P. 309–317.

20. Украина переведет все государственные данные на блокчейн URL: <https://hightech.fm/2017/04/14/us-ukraine-bitfury-blockchain> (дата звернення: 15.10.2019).

21. Pandia: Вибір криптографічно стійких еліптичних кривих URL: <http://pandia.ru/text/80/176/51044.php> (дата звернення: 18.10.2019).

22. Навчальні відео-матеріали від компанії Distributedlab URL: <https://distributedlab.com/> (дата звернення: 18.10.2019).

23. Через сутки вознаграждение за блок Bitcoin упадет вдвое URL: <https://habr.com/ru/post/395811/> (дата звернення: 03.10.2019).

24. Andreas M. Antonopoulos Mastering Біткоїн: Unlocking Digital Cryptocurrencies URL: <https://www.twirpx.com/file/61732/> (дата звернення: 22.10.2019).

25. Litecoin VS Ethereum. URL: <https://blockgeeks.com/guides/litecoin-vs-ethereum/?amazonai-language=en> (дата звернення: 18.10.2019).
26. Why are cryptocurrencies so volatile? URL: <https://www.quora.com/Why-are-cryptocurrencies-so-volatile.com/>
27. Метод ближайших соседей URL:  
[http://www.machinelearning.ru/wiki/index.php?title=Метод\\_ближайшего\\_соседа](http://www.machinelearning.ru/wiki/index.php?title=Метод_ближайшего_соседа)  
(дата звернення: 11.10.2019).
28. Giving a twitter bot ability to predict Біткоїн value based on historical data URL: <https://hackernoon.com/giving-a-twitter-bot-ability-to-predict-Біткоїн-value-based-on-historical-data-dbe237c40430> (дата звернення: 18.10.2019).
29. MIT researchers use Big Data to predict Bitcoin's value URL:  
<https://siliconangle.com/2014/10/22/mit-researchers-use-big-data-to-predict-bitcoins-value/> (дата звернення: 15.10.2019).
30. Shah D, Zhang K. Bayesian regression and Bitcoin. Proceedings of the 52nd Annual Allerton conference on communication, control, and computing, Allerton. 2014. P. 409–414.
31. Біткоїн price prediction algorithm using bayesian regression techniques URL: <https://github.com/panditanvita/BTSpredictor> (дата звернення: 20.10.2019).
32. Blockchain Deep Dive URL: <https://blockchainhub.net/blockchain-deep-dive/> (дата звернення: 02.10.2019).
33. Інновації втілені в реальних продуктах та послугах. Найцікавіші способи використання Біткоїн ще належить відкрити. URL:  
<https://bitcoin.org/uk/innovation> (дата звернення: 05.10.2019).
34. Самая мирная революция. URL: <https://habr.com/ru/post/480028/> (дата звернення: 09.10.2019).
35. Эстония поможет Украине построить "государство в смартфоне". URL: <https://delo.ua/economyandpoliticsinukraine/estonija-pomozhet-ukraine-postroit-gosudarstvo-v-358881/> (дата звернення: 15.10.2019).
36. Madan I., Saluja S., Zhao A. Automated Bitcoin Trading via Machine Learning Algorithms. Stanford, 2014. P. 2-5

37. Financial, Economic and Alternative Data. Quandl URL:  
<https://www.quandl.com/> (дата звернення: 22.10.2019)
38. Бідюк П.І., Коршевніук Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень. Київ, 2010. С. 115–136.
39. Fawcett, Tom. An Introduction to ROC Analysis. Pattern Recognition Letters. Elsevier, 2006. P. 7–12.
40. Bidyuk B. P. Exploiting graph cutsets for sampling-based approximations in Bayesian networks. PhD dissertation / University of California. Irvine, July 2006. 254p.
41. Бидюк П. И. Построение и методы обучения байесовских сетей. *Таврический вестник информатики и математики*. Симферополь: КНЦ НАНУ, 2004. № 2. С. 139–153.
42. Бидюк П. И. Метод вероятностного вывода в байесовских сетях по обучающим данным. *Кибернетика и системный анализ*. 2007. № 3. С. 93-99.
43. Згуровський М. З., Бідюк П.І., Терентьев О. М., Просянкін-ЖароваТ. І. Байєсівські мережі в системах підтримки прийняття рішень. Київ: ТОВ Видавниче Підприємство Едельвейс, 2015. С. 234–262.
44. Murphy K. A brief introduction to graphical models and Bayesian networks. Technical report / University of British Columbia. Canada, May 2001. 19 p.
45. Zweig G. G. Speech recognition with dynamic Bayesian networks. Proceedings of the fifteenth conference on artificial intelligence, Madison, Wisconsin US. 1998. P. 173-180.
46. Хабаров С. П. Экспертные системы URL:  
<https://www.twirpx.com/file/61731/> (дата звернення: 22.10.2019).
47. Spirtes, P., Glymour, C. Scheines, R., Causation, prediction and search. Adaptive computation and machine learning. MIT press, 2001. P. 523–565
48. Jouffe L. and Munteanu P. New search strategies for learning Bayesian networks. Proceedings of tenth international symposium on applied stochastic models and data analysis. ASMDA Compiègne, 12–15 June 2001. P. 591–596.

49. Бідюк П.І., Меньяйленко О.С., Половцев О. В. Методи прогнозування. Луганськ: Альма-Матер, 2008. С. 583–607.
50. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. SF.: Morgan Kaufmann, September 1988. P. 552
51. Guo H., Hsu W. A survey of algorithms for real-time Bayesian network inference. Proceedings of the AAAI-02/KDD-02/UAI-02 workshop on real-time decision support and diagnosis systems, Edmonton, Alberta, Canada, 1–4 August, 2002. SF.: Morgan Kaufmann, 2002. P. 1-12.
52. Pearl J. Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, NY.: Elsevier, September 1986. Vol. 29, №3. P. 241–288.
53. Zhang N. L., Poole D. A simple approach to Bayesian network computations. Proceedings of the Tenth Canadian Conference on Artificial Intelligence, 1994. P. 171-178.
54. Dechter R. Bucket elimination: a unifying framework for reasoning. ACM Press, December 1996. Vol. 28, № 61. P. 1-51.
55. SAS Institute. Step-By-Step Programming with Base SAS 9.4. SAS Institute, 2013. P. 712–725.
56. SAS® OnDemand for Academics. URL: [https://www.sas.com/en\\_us/software/on-demand-for-academics.html](https://www.sas.com/en_us/software/on-demand-for-academics.html) (дата звернення: 22.10.2019).
57. Python Release Python 3.8.0 URL: <https://www.python.org/downloads/release/python-380/> (дата звернення: 22.10.2019).
58. Graph Editor URL: [https://csacademy.com/app/graph\\_editor/](https://csacademy.com/app/graph_editor/) (дата звернення: 22.10.2019).

# ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

## А.1 Обробка вхідних даних

```
libname filedir "C:\work\23_11_2017";

PROC IMPORT OUT=filedir.dataset
            DATAFILE='C:\sas_rus.csv'
            DBMS=CSV replace;
            delimiter=';';
            GETNAMES=YES;
RUN;

/*PROC IMPORT OUT=filedir.dataset
    DATAFILE='C:\sas_rus.txt'
    DBMS=TAB REPLACE;
    GETNAMES=YES;
RUN;*/
options nosource nosource2 nonotes;
ODS RESULTS OFF;

proc contents
DATA=filedir.dataset
OUT=filedir.columns (keep=NAME) noprint;
run ;

%macro get_table_size(inset,macvar);
    data _null_;
        set &inset NOBS=size;
        call symput("&macvar",size);
    stop;
    run;
%mend;

%let var_count=;
%let iter_count=0;
%get_table_size(FILEDIR.columns,var_count);
%let var_count = &var_count.;
%let total_observations = 0;
%get_table_size(FILEDIR.dataset,total_observations);
%let total_observations = &total_observations.;
%let obs_count=;
%get_table_size(FILEDIR.GRAPH,obs_count);
%let obs_count = &obs_count.;
%let min =0;
%let putt =0;

PROC SQL noprint;
select NAME
into :Name1 - :Name&var_count.
from filedir.columns;
quit;

data filedir.MI(keep= _TYPE_);
```



```

set filedir.dataset;
IF _N_ = 1 THEN
    _TYPE_ = 0;
run;

DATA filedir.MI;
SET filedir.MI;
IF CMISS(OF _ALL_) THEN
    DELETE;
run;

DATA filedir.H;
SET filedir.MI;
run;

DATA filedir.ALPHA;
SET filedir.MI;
run;

DATA filedir.GRAPH;
SET filedir.MI;
run;

DATA filedir.EMPT;
SET filedir.MI;
run;

```

## A.2 Підрахунок імовірностей вершин

```

%MACRO DO_DIVIDE;
%DO I = 1 %TO &var_count.;

    DATA filedir.&&Name&I.;
    SET filedir.dataset(keep = &&Name&I.);
    cnt = 1;
    run;

    PROC SUMMARY DATA = filedir.&&Name&I.;
    CLASS &&Name&I.;
    VAR cnt;
    OUTPUT OUT = filedir.cnt_&&Name&I.
        SUM(cnt) = COUNT;
    run;

    DATA filedir.cnt_&&Name&I.;
    SET filedir.cnt_&&Name&I.;
    IF _N_ = 1 THEN
        SUM = COUNT;
    RETAIN;
    run;

    DATA filedir.cnt_&&Name&I.;
    SET filedir.cnt_&&Name&I.;
    PROB = COUNT/SUM;

```

```

run;

DATA filedir.cnt_&&Name&I.;
SET filedir.cnt_&&Name&I.;
IF CMISS(OF _ALL_) THEN
    DELETE;
run;

DATA filedir.cnt_&&Name&I.;
SET filedir.cnt_&&Name&I.;
    H = -COUNT * LOG(COUNT/SUM);
run;

PROC SUMMARY DATA = filedir.cnt_&&Name&I.;
VAR H;
OUTPUT OUT = filedir.H_&&Name&I.
    SUM(H) = H_&&Name&I.;
run;

DATA filedir.H (drop= _FREQ_);
    merge filedir.H filedir.H_&&Name&I.;
    by _TYPE_;
run;
%END;

%DO I = 1 %TO &var_count.;
    %DO K = 1 %TO &var_count.;
        %IF &I LT &K %THEN %DO;

            DATA filedir.&&Name&I.&&Name&K.;
            SET filedir.dataset(keep= &&Name&I. &&Name&K.);
                cnt = 1;
            run;

            PROC SUMMARY DATA = filedir.&&Name&I.&&Name&K.;
            CLASS &&Name&I. &&Name&K.;
            VAR cnt;
            OUTPUT OUT = filedir.cnt_&&Name&I.&&Name&K.
                SUM(cnt) = COUNT;
            run;

            DATA filedir.cnt_&&Name&I.&&Name&K.;
            SET filedir.cnt_&&Name&I.&&Name&K.;
                IF _N_ = 1 THEN
                    SUM = COUNT;
                RETAIN;
            run;

            DATA filedir.cnt_&&Name&I.&&Name&K.;
            SET filedir.cnt_&&Name&I.&&Name&K.;
                PROB = COUNT/SUM;
            run;

            DATA filedir.cnt_&&Name&I.;
            SET filedir.cnt_&&Name&I.;
            IF CMISS(OF _ALL_) THEN
                DELETE;
            run;
        %END;
    %END;

```

```

PROC SQL;
  CREATE TABLE filedir.QRY_&&Name&I.&&Name&K. AS
  SELECT t1.&&Name&I.,
         t1.&&Name&K.,
         t1.COUNT AS COUNT_&&Name&I.&&Name&K.,
         t1.SUM,
         t1.PROB AS PROB_&&Name&I.&&Name&K.,
         t2.COUNT AS COUNT_&&Name&I.,
         t2.PROB AS PROB_&&Name&I.
    FROM FILEDIR.CNT_&&Name&I.&&Name&K. t1
   INNER JOIN FILEDIR.CNT_&&Name&I. t2 ON
(t1.&&Name&I. = t2.&&Name&I.);
QUIT;

PROC SQL;
  CREATE TABLE filedir.CNT_&&Name&I.&&Name&K. AS
  SELECT t1.&&Name&I.,
         t1.&&Name&K.,
         t1.COUNT_&&Name&I.&&Name&K.,
         t1.SUM,
         t1.PROB_&&Name&I.&&Name&K.,
         t1.COUNT_&&Name&I.,
         t1.PROB_&&Name&I.,
         t2.COUNT AS COUNT_&&Name&K.,
         t2.PROB AS PROB_&&Name&K.
    FROM FILEDIR.QRY_&&Name&I.&&Name&K. t1
   INNER JOIN FILEDIR.CNT_&&Name&K. t2 ON
(t1.&&Name&K. = t2.&&Name&K.);
QUIT;

```

### A.3 Оцінка значень MI

```

DATA filedir.CNT_&&Name&I.&&Name&K.;
SET filedir.CNT_&&Name&I.&&Name&K.;
  MUT_INF_&&Name&I.&&Name&K. =
PROB_&&Name&I.&&Name&K. * ( LOG (PROB_&&Name&I.&&Name&K. /
                           PROB_&&Name&I. / PROB_&&Name&K.) );
run;

PROC SUMMARY DATA = filedir.cnt_&&Name&I.&&Name&K.;
VAR MUT_INF_&&Name&I.&&Name&K.;
OUTPUT OUT = filedir.MI_&&Name&I.&&Name&K.
      SUM(MUT_INF_&&Name&I.&&Name&K.) =
MI_&&Name&I.&&Name&K.;
run;

DATA filedir.MI (drop= _FREQ_);
  merge filedir.MI filedir.MI_&&Name&I.&&Name&K.;
  by _TYPE_;
run;

```

## A.4 Оцінка значень Н

```

DATA filedir.cnt_&&Name&I.&&Name&K.;
SET filedir.cnt_&&Name&I.&&Name&K.;
    Hhhh_&&Name&I.&&Name&K. = -COUNT_&&Name&I.&&Name&K.
*

LOG (COUNT_&&Name&I.&&Name&K./COUNT_&&Name&I.);
    Hhhh_&&Name&K.&&Name&I. = -COUNT_&&Name&I.&&Name&K.
*

LOG (COUNT_&&Name&I.&&Name&K./COUNT_&&Name&K.);
run;

PROC SUMMARY DATA = filedir.cnt_&&Name&I.&&Name&K.;
VAR Hhhh_&&Name&I.&&Name&K. Hhhh_&&Name&K.&&Name&I.;
OUTPUT OUT = filedir.H_&&Name&I.&&Name&K.
    SUM(Hhhh_&&Name&I.&&Name&K.) = H_&&Name&I.&&Name&K.
    SUM(Hhhh_&&Name&K.&&Name&I.) =
H_&&Name&K.&&Name&I.;
run;

DATA filedir.H (drop= _FREQ_);
merge filedir.H filedir.H_&&Name&I.&&Name&K.;
by _TYPE_;
run;

                                %END;
%END;

%END;
%MEND DO_DIVIDE;

%DO_DIVIDE;

```

## A.5 Ранжування вершин за значенням МІ

```

proc transpose data=filedir.MI
    out=filedir.MI_tr;
run;

proc transpose data=filedir.H
    out=filedir.H_tr;
run;

PROC SQL;
    CREATE TABLE filedir.MI_tr_ordr AS
    SELECT t1._NAME_,
           t1.COL1
    FROM FILEDIR.MI_TR t1
    ORDER BY t1.COL1 DESC;
QUIT;

```

```

PROC SQL;
  CREATE TABLE filedir.H_tr_ordr AS
  SELECT t1._NAME_,
         t1.COL1
  FROM FILEDIR.H_TR t1
  ORDER BY t1._NAME_ DESC;
QUIT;

```

## A.6 Оцінка значень $\alpha$

```

%MACRO DO_ALPHA;
  %DO I = 1 %TO &var_count.;
    PROC SUMMARY DATA = filedir.CNT_&&Name&I.;
      VAR _TYPE_;
      OUTPUT OUT = filedir.alpha_&&Name&I. (drop= _FREQ_)
        SUM(_TYPE_) = ALPHA_&&Name&I.;
    run;

    DATA filedir.ALPHA;
      merge filedir.ALPHA filedir.alpha_&&Name&I.;
      by _TYPE_;
    run;
  %END;

  DATA filedir.GRAPH;
  SET filedir.GRAPH;
  %DO I = 1 %TO &var_count.;
    %DO K = 1 %TO &var_count.;
      %IF &I LT &K %THEN %DO;
        ARC_&&Name&I.&&Name&K. = 0;
        %let iter_count = %eval(&iter_count. + 1);
      %END;
    %END;
  %END;

  _TYPE_ = 0;
  run;

  DATA filedir.GRAPH;
    merge filedir.GRAPH filedir.ALPHA;
    by _TYPE_;
  run;

  DATA filedir.GRAPH;
    merge filedir.GRAPH filedir.MI;
    by _TYPE_;
  run;

  DATA filedir.GRAPH;
    merge filedir.GRAPH filedir.H;
    by _TYPE_;
  run;
  %put &iter_count.;

```

## A.7 Побудова структур мереж

```

%DO ITERATION = 1 %TO &iter_count.;
%put &ITERATION.;
DATA _null_;
SET filedir.MI_tr_ordr;
IF _N_ = &ITERATION. THEN
    call symput("CURRENT_ARC", _NAME_);
run;

%DO I = 1 %TO &var_count.;
    %DO K = 1 %TO &var_count.;
        %IF &I LT &K %THEN %DO;
            %IF &CURRENT_ARC. = MI_&&Name&I.&&Name&K.
%THEN %DO;

DATA filedir.GRAPH_appendable;
SET filedir.GRAPH;
run;

%get_table_size(FILEDIR.GRAPH, obs_count);
%let obs_count=&obs_count.;

proc datasets;
    append base=filedir.GRAPH
data=filedir.GRAPH_appendable;
run;

proc datasets;
    append base=filedir.GRAPH
data=filedir.GRAPH_appendable;
run;

DATA filedir.GRAPH;
SET filedir.GRAPH;
IF _N_ LE &obs_count. THEN
    ARC_&&Name&I.&&Name&K. = 0;
ELSE IF _N_ LE (&obs_count. * 2) THEN
    ARC_&&Name&I.&&Name&K. = 1;
ELSE IF _N_ LE (&obs_count. * 3) THEN
    ARC_&&Name&I.&&Name&K. = -1;
run;

proc contents
DATA=filedir.GRAPH
OUT=filedir.columns_GRAPH (keep=NAME)
noprnt;

run ;

%get_table_size(FILEDIR.GRAPH, obs_count);
%let obs_count=&obs_count.;

```

## A.8 Оцінка ациклічності графа.

```

%DO OBSERVATION = 1 %TO &obs_count.;

DATA filedir.GRAPH;
SET filedir.GRAPH;
%DO START = 1 %TO &var_count.;
  V_&START. = 0;
  VV_&START. = 0;
  CYCLIC = 0;
%END;
%DO START = 1 %TO &var_count.;
  V_&START. = 1;
  %DO ITERR = 1 %TO

&var_count.;

                                %DO CHILD = 1 %TO

&var_count.;

                                VV_&CHILD. = 0;
                                %DO PARENT = 1 %TO

&var_count.;

                                IF
(V_&PARENT. = 1) AND (&CHILD. LT &PARENT.) AND
(ARC_&&Name&CHILD.&&Name&PARENT. = -1) THEN

                                VV_&CHILD. = 1;

                                IF
(V_&PARENT. = 1) AND (&CHILD. GT &PARENT.) AND
(ARC_&&Name&PARENT.&&Name&CHILD. = 1) THEN

                                VV_&CHILD. = 1;

                                %END;
                                %END;
                                %DO CHILD = 1 %TO

&var_count.;

                                V_&CHILD. =

VV_&CHILD.;

                                %END;
                                %END;
                                %DO CHILD = 1 %TO

&var_count.;

                                IF V_&CHILD. = 1 THEN
                                    CYCLIC = 1;
                                V_&CHILD = 0;
                                %END;
%END;
run;

DATA filedir.GRAPH_&OBSERVATION.;
SET filedir.GRAPH;
IF _N_ = 1 THEN
  OUTPUT;
run;

```

```

DATA filedir.GRAPH;
SET filedir.GRAPH;
IF _N_ = 1 THEN
    DELETE;
run;
%END;

%DO OBSERVATION = 1 %TO &obs_count.;
    proc datasets;
        append base=filedir.GRAPH
data=filedir.GRAPH_&OBSERVATION.;
    run;
%END;

DATA filedir.GRAPH;
SET filedir.GRAPH;
IF CYCLIC = 1 THEN
    DELETE;
run;

%get_table_size(FILEDIR.GRAPH,obs_count);
%let obs_count=&obs_count.;

%DO OBSERVATION = 1 %TO &obs_count.;

    DATA filedir.GRAPH;
    SET filedir.GRAPH;
    L = 0;
    %DO II = 1 %TO &var_count.;
        Hj=0;
        kj = ALPHA_&&Name&II. - 1;
        %DO KK = 1 %TO &var_count.;
            %IF &II LT &KK %THEN

%DO;
IF
ARC_&&Name&II.&&Name&KK. = -1 THEN
DO;
Hj = Hj +
kj = kj *

ALPHA_&&Name&KK.;
END;
%END;
%IF &II GT &KK %THEN

%DO;
IF
ARC_&&Name&KK.&&Name&II. = 1 THEN
DO;
Hj = Hj +
kj = kj *

ALPHA_&&Name&KK.;
END;
%END;
%END;
IF Hj = 0 THEN

```



```

                                Hj = H_&&Name&II.;
                                L_&&Name&II. = Hj + 1 / 2 *
kj * LOG(&total_observations.);

                                L = L + L_&&Name&II.;
                                %END;
                                run;

                                DATA filedir.GRAPH_&OBSERVATION.;
                                SET filedir.GRAPH;
                                IF _N_ = 1 THEN
                                    OUTPUT;
                                run;

                                DATA filedir.GRAPH;
                                SET filedir.GRAPH;
                                IF _N_ = 1 THEN
                                    DELETE;
                                run;
                                %END;

                                %DO OBSERVATION = 1 %TO &obs_count.;
                                proc datasets;
                                append base=filedir.GRAPH
data=filedir.GRAPH_&OBSERVATION.;
                                run;
                                %END;

                                PROC SQL;
                                CREATE TABLE filedir.GRAPH_ordr AS
                                SELECT *
                                FROM FILEDIR.GRAPH t1
                                ORDER BY t1.L;
                                QUIT;

                                DATA filedir.GRAPH_ordr;
                                SET filedir.GRAPH_ordr;
                                IF _N_ = 1 THEN
                                DO;
                                    MIN = L;
                                    retain min;
                                END;
                                run;

                                DATA filedir.GRAPH;
                                SET filedir.EMPT;
                                run;

                                DATA filedir.GRAPH (drop= MIN);
                                SET filedir.GRAPH_ordr;
                                WHERE L = MIN;
                                run;

                                DATA filedir.GRAPH;
                                SET filedir.GRAPH;
                                call symput("putt", L);
                                run;

                                %put &putt.;

```

```

%get_table_size(FILEDIR.GRAPH,obs_count);
                                %let obs_count=&obs_count.;
                                %put obs_count=&obs_count.;
                                %END;
                                %END;
                                %END;
                                %END;
                                %END;
%MEND DO_ALPHA;

%DO_ALPHA;

```

## A.9 Формування результатів

```

%MACRO DO_REZ;
    %get_table_size(FILEDIR.GRAPH,obs_count);
    %let obs_count=&obs_count.;
    %put &obs_count.;

%DO OBSERVATION = 1 %TO &obs_count.;
    DATA filedir.GRAPH_FIR;
    SET filedir.EMPT;
    run;

    DATA filedir.GRAPH_FIR;
    SET filedir.GRAPH;
    IF _N_ = &OBSERVATION. THEN
        OUTPUT;
    run;

    proc transpose data=filedir.GRAPH_FIR
        out=filedir.REZ_&OBSERVATION.;
    run;
    %put &OBSERVATION.;

    DATA filedir.REZ_&OBSERVATION.;
    SET filedir.REZ_&OBSERVATION.;
    IF ((COL1 NE 1) AND (COL1 NE -1)) THEN
        DELETE;
    run;

    DATA filedir.REZ_&OBSERVATION.;
    SET filedir.REZ_&OBSERVATION.;
    format c1 $char32.
           c2 $char32.;
    %DO I = 1 %TO &var_count.;
        %DO K = 1 %TO &var_count.;
            %IF &I LT &K %THEN %DO;
                IF _NAME_ = "ARC_&Name&I.&Name&K." THEN
                    DO;
                IF COL1 = 1 THEN
                    DO;

```

```

                                c1 = "&&Name&I.";
                                c2 = "&&Name&K.";
                                END;
                                IF COL1 = -1 THEN
                                    DO;
                                        c1 = "&&Name&K.";
                                        c2 = "&&Name&I.";
                                    END;
                                END;
                                END;
                                %END;
                                %END;
                                %END;
                                %END;
                                DATA filedir.REZ_&OBSERVATION.(keep= c1 c2);
                                SET filedir.REZ_&OBSERVATION.;
                                run;

                                ODS RESULTS ON;
                                proc print data=filedir.REZ_&OBSERVATION.;
                                run;
                                ODS RESULTS OFF;
                                %END;
                                %MEND DO_REZ;

                                %DO_REZ;

```

## A.10 Клас Net

```

class Net:
    def __init__(self, Bayesian_Net):
        self.list_of_factors = []
        self.evidence = {}
        self.var2eliminate = []
        for element in Bayesian_Net:
            self.evidence[element] = 5
            my_factor = Factor()
            my_factor._fillin_initial_element(element,
Bayesian_Net[element])
            self.list_of_factors.append(my_factor)
        def _set_evidence_net(self):
            no_queried_variables = True
            for element in self.evidence:
                if self.evidence[element] == 0 or self.evidence[element] ==
1:
                    for factor in self.list_of_factors[:]:
                        factor._set_evidence(element,
self.evidence[element])
                        if len(factor.elements) == 0:
                            self.list_of_factors.remove(factor)
                        elif self.evidence[element] == 2:
                            self.var2eliminate.append(element)
                        else:
                            no_queried_variables = False
            if no_queried_variables:
                raise NameError('No queried variables!')
        def _calculate_cost(self, element):
            dependent_elements = []
            for factor in self.list_of_factors:

```

```

        if element in factor.elements:
            for element_ in factor.elements:
                if ((element_ not in dependent_elements)
                    and element_ != element):
                    dependent_elements.append(element_)
    return len(dependent_elements)
def _choose_var2eliminate(self):
    try:
        next_var2eliminate = self.var2eliminate[0]
        cost_var2eliminate =
self._calculate_cost(next_var2eliminate)
        for element in self.var2eliminate:
            aux_cost_var2eliminate = self._calculate_cost(element)
            if aux_cost_var2eliminate < cost_var2eliminate:
                next_var2eliminate = element
                cost_var2eliminate = aux_cost_var2eliminate
        return next_var2eliminate
    except:
        return None
def _eliminate_variable(self, element):
    factors_involved = []
    for factor in self.list_of_factors:
        if element in factor.elements:
            factors_involved.append(factor)
    new_factor = Factor()
    new_factor._build_factor(factors_involved)
    new_factor._marginalize(element)
    for factor in factors_involved[:]:
        self.list_of_factors.remove(factor)
    if len(new_factor.elements) > 0:
        self.list_of_factors.append(new_factor)
    self.var2eliminate.remove(element)

def _mult_factors(self):
    new_factor = Factor()
    new_factor._build_factor(self.list_of_factors)
    for factor in self.list_of_factors[:]:
        self.list_of_factors.remove(factor)
    self.list_of_factors.append(new_factor)

def _variable_elimination(self):
    self._set_evidence_net()

    while True:
        var2eliminate = self._choose_var2eliminate()
        if var2eliminate == None:
            break
        self._eliminate_variable(var2eliminate)
    if len(self.list_of_factors) > 1:
        self._mult_factors()
    self.list_of_factors[0]._normalize()

```

## ДОДАТОК Б РОБОЧИЙ ДАТАСЕТ

Date	Diff MKPRU	Diff MWNUS	Diff TRVOU	Diff MWNTD	Diff MWTRV	Diff AVBLS	Diff BLCHS
15.06.2018	1	20423,36	-67632615,2	-3396,478	-28894,5478	-0,18685298	113,7356453
16.06.2018	-1	20423,36	-331321060	1585,572	-7665,71677	0,045883188	122,4653468
17.06.2018	1	20423,36	-67632615,2	-19768,99	-28894,5478	-0,18685298	78,3539874
18.06.2018	-1	11170,84	-67632615,2	-3396,478	-2285,97929	0,021366215	78,3539874
19.06.2018	1	17340,69	352639367,2	22587,1	27090,70117	0,196298169	102,6313637
20.06.2018	1	20423,36	-67632615,2	-1352,722	-2285,97929	0,196298169	141,6504434
21.06.2018	-1	17340,69	59142176,86	1585,572	-2285,97929	0,196298169	161,1259307
22.06.2018	1	27690,49	-67632615,2	1585,572	7029,578184	-0,18685298	102,6313637
23.06.2018	-1	20423,36	352639367,2	-3396,478	1349,132423	0,196298169	122,4653468
24.06.2018	-1	35689,17	59142176,86	-5795,594	-7665,71677	-0,18685298	78,3539874
25.06.2018	-1	35689,17	-331321060	-3396,478	-2285,97929	-0,04972048	78,3539874
26.06.2018	1	27690,49	59142176,86	22587,1	27090,70117	0,196298169	141,6504434
27.06.2018	1	27690,49	-331321060	-1352,722	1349,132423	0,196298169	161,1259307
28.06.2018	-1	35689,17	-27297505,6	-1352,722	7029,578184	0,021366215	132,3506878
29.06.2018	1	27690,49	-67632615,2	-1352,722	7029,578184	-0,18685298	102,6313637
30.06.2018	-1	67638,99	352639367,2	-37,72222	-7665,71677	0,045883188	113,7356453
01.07.2018	1	67638,99	59142176,86	-3396,478	-7665,71677	0,196298169	141,6504434
02.07.2018	-1	20423,36	-331321060	-9158,433	-7665,71677	-0,18685298	102,6313637
03.07.2018	1	35689,17	59142176,86	10596,16	27090,70117	0,196298169	141,6504434
04.07.2018	1	67638,99	59142176,86	1585,572	7029,578184	0,196298169	132,3506878
05.07.2018	-1	35689,17	-8789040,88	-1352,722	1349,132423	-0,18685298	122,4653468
06.07.2018	1	67638,99	17333184,51	-37,72222	-7665,71677	-0,18685298	102,6313637
07.07.2018	-1	35689,17	-67632615,2	-37,72222	1349,132423	-0,08795593	113,7356453
08.07.2018	1	67638,99	-67632615,2	-9158,433	-28894,5478	-0,18685298	78,3539874
09.07.2018	1	35689,17	59142176,86	-3396,478	-2285,97929	0,045883188	78,3539874
10.07.2018	-1	35689,17	-67632615,2	22587,1	27090,70117	0,196298169	141,6504434
11.07.2018	-1	35689,17	352639367,2	-37,72222	255985,1276	-0,08795593	141,6504434
12.07.2018	-1	35689,17	-27297505,6	-3396,478	-256055,905	0,045883188	132,3506878
13.07.2018	-1	27690,49	-67632615,2	-1352,722	1349,132423	0,196298169	161,1259307
14.07.2018	1	35689,17	-8789040,88	1585,572	1349,132423	-0,08795593	161,1259307
15.07.2018	-1	27690,49	-67632615,2	-9158,433	-28894,5478	0,045883188	150,1311008

## ДОДАТОК В ЗАЯВКА НА АВТОРСЬКЕ СВІДОЦТВО

B1

МІНЕКОНОМРОЗВИТКУ

**ДЕРЖАВНЕ ПІДПРИЄМСТВО  
"УКРАЇНСЬКИЙ ІНСТИТУТ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ"  
(УКРПАТЕНТ)**

вул. Глазунова, 1, м. Київ-42, 01601, Україна Тел.: (044) 494-05-05 Факс: (044) 494-05-06  
E-mail: office@ukrpatent.org

01.08.2019 № 19673/ЗУ/19

Стосується заявки № у 2019 08349  
/ при листуванні просимо посилатися на цей № /

Адреса для листування  
Просянкін-Жарова Тетяна Іванівна,  
вул. Шевченка, буд. 33, кв. 22, м.  
Умань, Черкаська обл., 20300

**Повідомлення щодо встановлення дати подання заявки на винахід (корисну модель)**

**(21) Реєстраційний номер заявки у 2019 08349**

**(22) Дата подання 16.07.2019**

**(71) Заявник(и)**

**МАКОГОН РОМАН ОЛЕКСАНДРОВИЧ, ТЕРЕНТЬЄВ ОЛЕКСАНДР  
МИКОЛАЙОВИЧ, ПРОСЯНКІНА-ЖАРОВА ТЕТЯНА ІВАНІВНА**

**(54) Назва винаходу /корисної моделі/**

**СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ З ВИЗНАЧЕННЯ  
КРЕДИТОСПРОМОЖНОСТІ ФІЗИЧНИХ ОСІБ ІЗ ВИКОРИСТАННЯМ МЕРЕЖ  
БАЙЄСА**

Повідомляємо, що відповідно до частини 11 статті 12 Закону України "Про охорону прав на винаходи і корисні моделі" (далі - Закон) та Порядку сплати зборів за дії, пов'язані з охороною прав на об'єкти інтелектуальної власності, затвердженого постановою Кабінету Міністрів України від 23.12.2004 р. № 1716 із останніми змінами внесеними постановою Кабінету Міністрів України від 12.06.2019 № 496 (далі - Порядок сплати), Вам необхідно:

- **подати документ про сплату збору (за кодом 10100) за подання заявки у розмірі 1920.00 UAH**

Документ про сплату (доплату) збору за подання заявки повинен надійти до Державного підприємства "Український інститут інтелектуальної власності" протягом 2 місяців від дати подання заявки. Цей строк продовжується, але не більш ніж на 6 місяців, якщо до його спливу, а саме протягом 2 місяців від дати подання заявки, буде подано відповідне клопотання та сплачено збір за його подання.

Відповідно до Порядку збір за подання клопотання про продовження строку надходження документа про сплату збору за подання заявки (за кодом 10200) сплачується у розмірі 800 гривень або із врахуванням першого абзацу п. 3 Порядку сплати - у розмірі 80 (на винахід / топографію інтегральних мікросхем) або у розмірі 160 (на корисну модель) гривень, або із врахуванням п. 3<sup>1</sup> або п. 3<sup>2</sup> Порядку - у розмірі 160 (на винахід / топографію інтегральних мікросхем) або 320 (на корисну модель) гривень.

У разі порушення вимог частини 11 статті 12 цього Закону рішення про встановлення дати подання заявки не надсилається, а заявка вважається відкликаною.

Нагадуємо, що збори сплачуються за такими реквізитами:

(у гривнях) Одержувач: ДП "Український інститут інтелектуальної власності"

Банк одержувача: АТ "Укресімбанк" м. Києва

Р/р 26008020020371



u201908349

2

Код банку 322313  
Код ЗКПО 31032378

**Шановні заявники!** У надісланій заяві про видачу патенту не вказані відомості про винахідників (код 72). Просимо внести винахідників до бланку заяви.

Начальник відділу



Л.М. Луценко

Виконавець  
Телефон

Яворська Л.Г.  
494-05-98